

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук


КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

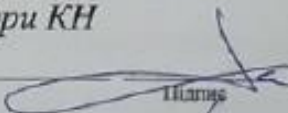
на тему Метод підвищення точності визначення зміщення об'єктів за
аналізом зображень з використанням методів кластеризації

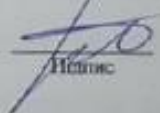
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності


Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 2 курсу, група КНм-22-1  Нікіта ВУСАТИЙ
Курс, група виконавця Підпис Ім'я, прізвище

Керівник: к.т.н., доцент кафедри КН  Олександр ПАСІЧНИК
Науковий ступінь, посада Підпис Ім'я, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, прізвище

09 грудня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

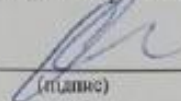
Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор Олександр БАРМАК

«02» Вересень 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації»

2. Завдання видано студенту Нікімі ВУСАТОМУ

(ім'я, прізвище)

3. Керівник роботи доцент кафедри КН, Олександр ПАСІЧНИК

(ім'я, прізвище)

4. Затверджені наказом університету від «26» Вересня 2024 р. №60.

5. Зміст пояснювальної записки (перелік задач) та вихідні дані: Наведено характеристику предметної області з оглядом методів аналізу зображень для визначення положення об'єктів, оглядом методів кластеризації для аналізу зображень та аналізом існуючих підходів до підвищення точності вимірювань у технічних системах. Визначено мету та задачі дослідження. Реалізовано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації та його програмна реалізація. Виконано тестування програмної реалізації та дослідження реалізованого методу.

Реферат

Кваліфікаційна робота магістра присвячена методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації.

Актуальність теми. Сьогодні неможливо уявити світ без інформаційних технологій. Куди не подивись, вони всюди використовуються. Починаючи від простої, але важливої автоматизації якихось процесів і закінчуючи дійсно складними технологіями з важкими розрахункам. Аналіз зображення стає все більш популярним у різноманітних сферах діяльності, у тому числі й при виявленні зміщення об'єктів спостереження та при визначенні величини цього зміщення.

Мета і задачі роботи. Метою роботи є підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. Для досягнення цієї мети були поставлені такі завдання:

- провести аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектувати метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначити набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконати програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначити точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.

Об'єкт дослідження. Процес визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації.

Предмет дослідження. Методи кластеризації.

Методи дослідження. Методи кластеризації, алгоритми обробки зображень.

Наукова новизна одержаних результатів. Удосконалено метод підвищення точності визначення зміщення об'єктів за аналізом зображень за рахунок використання кластеризації при визначенні контурів маркера на рухомому об'єкті.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук (АПКН – 20243)”, м. Хмельницький, ХНУ, 15-16 листопада 2024 р. (Вусатий Н.О., Пасічник О.А., Скрипник Т.К. Сегментація зображень // Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції “Актуальні проблеми комп’ютерних наук (АПКН – 2024)”. – Хмельницький: ХНУ, 2024. – С. 112 – 114. <https://elar.khmnu.edu.ua/handle/123456789/171373>).

За темою кваліфікаційної роботи магістра автором виконано 1 наукову публікацію - Вусатий Н., Пасічник О., Манзюк Е., Скрипник Т. Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах // Вісник Хмельницького національного університету, №4, 2024 (339), С. 59 – 63. DOI 10.31891/2307-5732-2024-339-4-10.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з вступу, чотирьох розділів, висновків, переліку посилань та додатків. Загальний обсяг роботи становить 133 сторінок, з яких 84 сторінок основного тексту, і включає 50 рисунків та 19 таблиці.

Зміст

Перелік скорочень	4
Вступ	5
РОЗДІЛ 1 Характеристика предметної області та постановка задачі	7
1.1 Огляд методів аналізу зображень для визначення положення об'єктів	7
1.2 Огляд методів кластеризації для аналізу зображень	14
1.3 Аналіз існуючих підходів до підвищення точності вимірювань у технічних системах	19
1.4 Висновки по розділу 1 та постановка задачі	21
РОЗДІЛ 2 Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації	22
2.1 Загальна структура методу кластеризації для аналізу зображень	22
2.2 Адаптація методу k-середніх	28
2.3 Кроки реалізації методу з використанням кластеризації	29
2.4 Математичне забезпечення методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації	34
2.5 Критерії покращення визначення зміщення	37
Висновки до розділу 2	39
РОЗДІЛ 3 Програмна реалізація методу підвищення точності визначення зміщення об'єктів	40
3.1 Архітектура та модулі програмної реалізації	40
3.2 Особливості програмної реалізації методу з кластеризацією	46
3.3 Функціональні можливості системи для аналізу зміщення об'єктів	47
3.4 Тестування програмної реалізації методу	48
Висновки до розділу 3	58
РОЗДІЛ 4 Дослідження методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації	59
4.1 Оцінка точності визначення зміщення об'єктів	59
4.2 Порівняння розробленого методу з іншими методами вимірювання	61
Висновки до розділу 4	78

Загальні висновки.....	80
Перелік посилань.....	82
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
K-means	Алгоритм кластеризації на основі методу середніх
LBP	Local Binary Patterns
ROI	Region of Interest
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
OpenCV	Open Source Computer Vision Library
YOLO	You Only Look Once
ADAS	Advanced Driver Assistance Systems
CNN	Convolutional Neural Networks
SVM	Support Vector Machine
ВК	Високий контраст
НК	Низький контраст

Вступ

Актуальність теми. Сьогодні неможливо уявити світ без інформаційних технологій. Куди не подивись, вони всюди використовуються. Починаючи від простої, але важливої автоматизації якихось процесів і закінчуючи дійсно складними технологіями з важкими розрахункам. Аналіз зображення стає все більш популярним у різноманітних сферах діяльності, у тому числі й при виявленні зміщення об'єктів спостереження та при визначенні, за потреби, величини цього зміщення.

Мета і задачі роботи. Метою роботи є підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. Для досягнення цієї мети були поставлені такі завдання:

- провести аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектувати метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначити набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконати програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначити точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.

Об'єкт дослідження. Процес визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації.

Предмет дослідження. Методи кластеризації.

Методи дослідження. Методи кластеризації, алгоритми обробки зображень.

Наукова новизна одержаних результатів. Удосконалено метод підвищення точності визначення зміщення об'єктів за аналізом зображень за рахунок використання кластеризації при визначенні контурів маркера на рухомому об'єкті.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук (АПКН – 20243)”, м. Хмельницький, ХНУ, 15-16 листопада 2024 р. (Вусатий Н.О., Пасічник О.А., Скрипник Т.К. Сегментація зображень // Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції “Актуальні проблеми комп’ютерних наук (АПКН – 2024)”. – Хмельницький: ХНУ, 2024. – С. 112 – 114. <https://elar.khmnu.edu.ua/handle/123456789/171373>).

За темою кваліфікаційної роботи магістра автором виконано 1 наукову публікацію - Вусатий Н., Пасічник О., Манзюк Е., Скрипник Т. Спосіб оцінки зміщення об’єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах // Вісник Хмельницького національного університету, №4, 2024 (339), С. 59 – 63. DOI 10.31891/2307-5732-2024-339-4-10.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з вступу, чотирьох розділів, висновків, переліку посилань та додатків. Загальний обсяг роботи становить 133 сторінок, з яких 84 сторінок основного тексту, і включає 50 рисунків та 19 таблиці.

РОЗДІЛ 1 Характеристика предметної області та постановка задачі

1.1 Огляд методів аналізу зображень для визначення положення об'єктів

Сьогодні неможливо уявити світ без інформаційних технологій. Куди не подивись, вони всюди використовуються. Починаючи від простої, але важливої автоматизації якихось процесів і закінчуючи дійсно складними технологіями з важкими розрахункам. І як не дивно, аналіз зображення стає все більш популярним у всіх сферах діяльності. У відеоспостереженні аналіз зображення допомагає виявити рухові об'єкти, наприклад під час руху автомобілів на трасах. Наша нинішня система працює таким чином, що відеокамера знімає весь рух, визначає об'єкт, його переміщення, його швидкість після чого робить висновок, чи перевищує швидкісний рух визначений автомобіль. І якщо порушує, визначає номер автомобіля та надсилає штраф в електронну систему. Це дійсно зручніше, ані ж якби людина особисто наводила на кожну машину спеціальним пристроєм, щоб визначити його швидкість. У сфері медицини аналіз зображення грає надважливу роль, наприклад для діагностування захворювання через розпізнання патології або під час планування хірургічних операцій. Сьогоднішня медицина існує за допомогою інформаційних систем, в тому числі і аналізу зображень. У робототехніці надважливо правильно, швидко і якісно проводити аналіз зображень, візьмемо для прикладу машини з автопілотом. У цьому випадку помилка системи може коштувати надто дорого. Тут необхідно постійно аналізувати зображення та робити правильні визначення. В цілому, люба робототехніка сьогодні працює на основі аналізу зображень, після чого вже йдуть інші дії. Наприклад автоматичний пилосос, який рушає по квартирі і прибирає, йому необхідно правильно аналізувати зображення, де він знаходиться, куди рухається, де він вже був, де вже прибрано, де неможливо проїхати і необхідно об'їхати та інше. І кажучи про автоматизацію, для прикладу це може бути будь-який веб-сайт, зазвичай на ньому немає ніяких складних обрахунків, проте його зручність, простота дуже прискорюють ручну роботу. Наприклад електронні черги будь-куди, пошук інформації, певні калькулятори для розрахунку калорію та інші

автоматизовані системи. Тобто куди не глянь, вже всюди застосовані інформаційні технології, та аналіз зображення є не менш важливим у їх роботі.

Кожен аналіз зображення починається з визначення положення об'єктів на зображенні. Відповідно, існує багато різних методів, усі вони мають свої переваги та недоліки, простіше кажучи, кожен метод призначений для конкретного використання, тобто майже не усіє якогось універсального методу для всіх задач.

Розглянемо найпопулярніші методи виявлення об'єктів за допомогою ключових точок та локальних ознак. Це два методи, перший SIFT [1] та другий SURF [2]. Їхня головна особливість і перевага полягає в тому, що вони шукають такі ознаки об'єктів, які не змінюються при зміні освітлення, масштабу, шуму, обертання та інших маніпуляцій. І це насправді дійсно дуже зручно, ці методи використовуються у багатьох галузях науки та мають справедливу популярність.

Власне, засновником методу SIFT вважається Девід Лоу. Метод має декілька етапів. Спочатку метод шукає спеціальні важливі точки змінюючи масштаб зображення та використовуючи на кожному зображенні фільтра Гауса [3]. Знайдені точки зберігаються та локалізуються для подальшого використання. Потім описується кожна знайдена точка, це може бути інформація про її положення, про її орієнтацію, розподіл інтенсивності або інша. Цей метод достатньо чітко і якісно визначає положення об'єктів, проте все-таки він є досить обчислювальним, і якщо для 10-20 зображень це не складає якихось проблем, то для аналізу зображень у реальному часі, це складно. Наприклад для відеокамери, що спостерігає рух автомобілів, це достатньо сильне навантаження. Також метод використовується в системах розпізнавання обличь, в тому числі з різних сторін, під різними кутами, освітлені та шуму.

Переваги SIFT:

- висока точність до змін масштабу, обертання та освітлення;
- поширене використання у робототехніці, медицини та інших наукових галузях;
- здатність ідентифікувати об'єкти в різних умовах зйомки.

Недоліки SIFT:

- досить високі обчислювальні витрати та велике навантаження на систему;
- має ліцензійне обмеження у використанні в комерційних цілях.

Наступний метод, метод SURF, засновник якого є Ерік Шоу, по суті, дуже схожий до SIFT, але вважається більш оптимізованим. Він має такі самі етапи та алгоритм як SIFT, але головна відмінність у тому, що він використовує фільтр Гауса як похідну другорядного порядку. Ця відмінність перекриває недолік методу SIFT, тобто дає можливість виконувати складні розрахунки достатньо швидко, при тому не зменшуючи точність розрахунків. Але це було занадто добре, якби все так легко працювало, тому він має свої недоліки, головний з них полягає у тому, що цьому методу важко працювати при змінних умовах зйомки, в такому випадку метод стає досить не точним у виявленні положення об'єктів.

Переваги SURF:

- не високі обчислювальні витрати у порівнянні з SIFT;
- зберігається висока точність обчислювання;
- можна використовувати для задач, які потребують спостереження у реальному часі завдяки оптимізації алгоритму.

Недоліки SURF:

- зменшення точності виявлення об'єктів при різних умовах освітлення та зйомки;
- не підходить для задач, де виявлення об'єктів повинно бути максимально точним.

Ще одним з способів виявлення об'єктів є виявлення їх країв. Простими словами, це спосіб, коли визначаються межі об'єкта. Найпопулярнішим серед них є алгоритм Кенні [4]. Щодо процесу роботи, він має наступні етапи:

1. Зображення згладжується за допомогою фільтра Гауса, щоб зменшити шум.
2. Обчислюється градієнт інтенсивності пікселів, з якого можна отримати напрямок і величину зміщення.

3. Потім застосовується немаксимальне придушення для точного налаштування високо градієнтних розташувань.
4. Застосовується подвійна порогова фільтрація для визначення справжніх країв.

Переваги алгоритму Кенні:

- досить висока точність виявлення контурів об'єктів;
- здатність виявляти тонкі та складні контури об'єктів;
- здатність зберігати точність виявлення при роботі з різними рівнями шумів.

Недоліки алгоритму Кенні:

- високі обчислювальні витрати.

Також існують відомі методи Собеля [5] та Превітта [6]. Вони засновані на застосуванні фільтра для розрахунку градієнта інтенсивності як у вертикальному, так і в горизонтальному напрямку. Вони досить прості у застосуванні та були визнані ефективними для основних завдань виявлення країв. Однак, порівняно з алгоритмом Кенні, вони погано працюють при різних налаштуваннях шумів.

Переваги методів Собеля та Превітта:

- достатньо простий алгоритм, простий у розуміння та реалізації;
- дуже ефективний для базових задач, які не потребують високої точності;
- має низькі обчислювальні витрати у порівнянні з іншими методами.

Недоліки методів Собеля та Превітта:

- не стійкий до зміни рівня шумів;
- не точність виявлення об'єктів при складних умовах.

Шаблонне зіставлення [7] також дуже поширений у використанні метод. Наприклад, якщо потрібно виявити певний об'єкт на зображенні, створюється шаблон цього об'єкта і знаходиться його на зображенні за допомогою кореляційних коефіцієнтів [8], це працює за допомогою пошуку частин зображення. Цей метод дійсно є дуже простим у використанні, у розумінні, у роботі, проте якщо змінити масштаб зображення, змінити освітлення, положення, рух, тоді такий метод не буде

працювати. Тому цей метод ідеально підходить для нерухомих об'єктів, для зображень, які є статичними і мають мінімальні зміни.

Процес роботи наведено нижче:

1. Спочатку обрається частина зображення, яка використовується як шаблон.
2. Після чого береться обраний шаблон зображення та порівнюється між шаблонами на кожній позиції за допомогою кореляційного коефіцієнта.
3. На останок отримуємо коефіцієнти та обираємо найвищі позиції з відповідними місцями для шаблону.

Переваги шаблонного зіставлення:

- достатньо простий у розумінні, реалізації та використанні;
- дуже ефективний в задачах стабільних зображень;
- використовується у системах розпізнання тексту.

Недоліки шаблонного зіставлення:

- погана точність роботи при різних рівнів масштабу, освітлення, шумів та в цілому складних умовах зйомки;
- досить висока обчислювальні витрати для великих та важких зображеннях, оскільки це створює досить багато шаблонів.

Проте, не зважаючи на недоліки, метод шаблонного зіставлення набув свою популярність і використовуються для контролю якості виробів на виробництві, оскільки там не має великих зображень, використовується для розпізнання тексту на зображеннях та у інших схожих галузях.

Методи машинного навчання, так звані згорткові нейронні мережі CNN [9]. На сьогодні нейронні мережі мають все більший попит, все більше популярність у використанні, у розробці. Кожного року системи з нейронними мережами стають все більше якісними і кращими. І на основі цього з'явився цей метод для аналізу зображень. Метод автоматично виділяє важливі ознаки зображення через послідовні шари згортки. Наприклад, мережі такі як VGG [10], ResNet [11] та Inception [12] показали надвисокі результати у багатьох тестуваннях. Тому що такі мережі здатні визначати та обробляти зображення зі складними аспектами та структурою.

Переваги CNN:

- достатньо висока точність у розпізнанні та класифікації об'єктів;
- дуже поширений у використанні, починаючи від медицини і закінчуючи автономний транспортних засобів.

Недоліки CNN:

- досить високі обчислювальні витрати, тому не підходить для роботи у реальному часі;
- чим більше даних, тим краще працює, чим менше даних, тим гірше працює, тобто метод має мати великий обсяг даних, аби показувати гарний результат;
- через залежність від даних, метод достатньо просто вразити не потрібними даними.

Коли мова йде про виявлення об'єктів у реальному часу, спершу це йде мова про алгоритм YOLO [13]. Його основна суть полягає у тому, що він ділить зображення на сітки і у той же самий час передбачає рух об'єктів та їх координати. Саме цей метод вигідно використовувати у багатьох ситуаціях, коло необхідно слідкувати та передбачати рух об'єктів на зображеннях.

Переваги YOLO:

- дуже висока швидкість роботи, особливо після останніх версій;
- здатність одночасно виявляти та локалізувати об'єкти;
- використовується у дуже багатьох галузях, через що має велику спільноту підтримки.

Недоліки YOLO:

- не підходить для задач, у яких важлива висока точність виявлення об'єктів.

Mask R-CNN [14] розширює алгоритм Faster R-CNN [15], додатково генеруючи маски сегментації для кожного виявленого об'єкта. Після чого визначаються їх межі. Зазвичай це використовується для складних сцен з перекриттями об'єктів або нестандартними формами.

Переваги Mask R-CNN:

- достатньо висока точність у сегментації та локалізації об'єктів;
- здатність працювати з різними та складними за формами об'єктами;
- зазвичай використовується у медицині для сегментації органів та патології, тобто для достатньо серйозних речей, тобто метод має високу репутацію.

Недоліки Mask R-CNN:

- високі обчислювальні витрати, тобто метод не підходить для роботи у реальному часі;
- складний для початкового розуміння та налаштування, проте потрібно лише один раз налаштувати та розібратися.

SVM [16], метод заснований на глибокому навчанні [32], використовується для класифікації та регресії об'єктів [17] на зображеннях. Зазвичай такий метод буває корисний для задач з чіткими розмежуваннями між класами. Він шукає оптимальну гіперплощину, яка повинна розділяти класи.

Переваги SVM:

- досить висока точність класифікації;
- метод ефективно працює навіть при великій кількості ознак;
- використовується у розпізнаванні образів, біометрії та у інших галузях науки.

Недоліки:

- має низьку ефективність роботи при малій кількості даних та класів, тобто чим менше даних, тим не точніше працює метод;
- не ефективний при наявності різних рівнів шумів;
- складний у налаштуванні.

Random Forest [18], метод, який в своїй основі використовує ансамбль рішень дерев для класифікації та регресії. Скажімо, метод призначений для роботи з великою кількістю ознак і працює достатньо чітко. Навіть при тому, коли метод не має багато даних для роботи, працює метод все-одно чітко і ефективно. Він зазвичай використовується для класифікації об'єктів у біологічних зображень та для розпізнавання текстур.

Переваги Random Forest:

- висока точність та можливість перенавчатися під нові дані;
- здатність не втратити високу точність при великому навантаженні даними;
- легкість у розумінні та використанні;
- використовується у досить серйозних галузях, що дає високу репутацію методу.

Недоліки Random Forest:

- високі обчислювальні витрати.

Методи машинного навчання зазвичай використовують у медичній діагностиці, для автономії транспортних засобів, розпізнаванні структур, біометрії та інших галузях науки.

1.2 Огляд методів кластеризації для аналізу зображень

У контексті аналізу зображень кластеризація допомагає поділити зображення на окремі області з подібними характеристиками, такими як колір, текстура або інтенсивність.

Існує багато методів кластеризації та кожен з них має свої особливості та підходить для певних завдань [19]. Найбільш популярними методами є k-середніх [20, 31], DBSCAN [21], ієрархічна кластеризація [22] та спектральна кластеризація [23].

Метод k-середніх є найбільш популярним, простим у використанні та розумінні. Він працює таким чином, що ділить зображення на k маркерів, де кожен маркер має свій центр, тобто так званий центроїд. Центроїд - це середня точка маркера серед усіх пікселів в маркері. І такий процес поділу відбувається постійно до того моменту, як маркер не стануть стабільними.

Процес роботи методу k-середніх:

1. Спочатку обирається кількість маркерів, на яку буде поділено дані, хоча іноді це робить автоматично за додатковим алгоритмом.

2. Далі випадковим чином обираються вказані k маркерів з даних, як початкові центри маркерів.
3. Кожна точка даних призначається до найближчого центру маркеру.
4. Після призначення усіх точок до маркерів, обчислюється новий центр кожного маркера як середнє значення усіх точок.
5. І так процес повторюється від 3 до 4 пункту до того моменту, доки центри маркерів не почнуть збігатися.

Переваги k -середніх:

- простота реалізації, простота в розумінні роботи метода, точність обробки;
- точність обробки при великому обсязі даних;
- швидке адаптування до нових даних, навіть з великим обсягом даних зображень або відео.

Недоліки k -середніх:

- необхідність заздалегідь вказувати кількість маркерів, проте вже існують алгоритми, які автоматично визначають цю кількість;
- не підходить для маркерів дуже складної форми.

Зазвичай використовується у задачах сегментації зображень, аналізу структур, аналізу геометричних фігур та біометричних даних.

DBSCAN одним з популярним методів кластеризації, проте іронічно те, що він не потребує заздалегідь визначеної кількості маркерів. Замість цього, він базується на основі щільності даних, відповідно цей простий спосіб дозволяє йому автоматично визначати кількість маркерів на основі шумових пікселів.

Процес роботи DBSCAN:

1. Вибираються два параметри, це радіус сусідства та мінімальна кількість точок для формування маркера.
2. Для кожної точки визначається чи є у її радіусі мінімальна кількість точок, яка була обрана.
3. І точки які мають достатню щільність сусідства, вони формують так зване ядро.

4. До кожного ядра приєднуються усі точки, які знаходяться в цьому радіусі сусідства.
5. А точки які не належать жодному маркеру, відносяться до шуму.

Переваги DBSCAN:

- не потребує заздалегідь вказаного кількості маркерів;
- здатність виявляти маркер будь-якої форми;
- ефективно працює з зображеннями різного рівня шуму.

Недоліки DBSCAN:

- не ефективний у задачах, де може бути різні щільність;
- потребує параметру радіусу та параметру мінімальної кількості точок;
- не ефективний у задачах високо розмірних даних.

Зазвичай цей метод використовується у медичних зображеннях, в аналізі географічних даних та у промисловому контролі.

Ієрархічна кластеризація, це такий вид кластеризація, який має ієрархію маркерів, де кожен маркер може бути розділений на під маркери або об'єднаний з іншим маркером. В цілому, існує два види ієрархічних методів кластеризації, агломератні та дивізійні.

Власне, агломератна кластеризація починається з кожної точки, як окремого маркера та поступово об'єднує найближчі маркери до тих пір, доки усі точки не будуть включені до одного великого маркера або в іншому випадку, доки не буде досягнуто тої кількості маркерів, яку вказав користувач.

Процес агломератної кластеризації:

1. Ініціалізація.
2. Об'єднання маркерів.
3. Створення ієрархії.

Дивізійна кластеризація працює таким чином, що має один великий маркер, який поступово ділиться на менші, і так відбувається до тих пір, доки кожен піксель не буде окремим маркером.

Переваги ієрархічної кластеризації:

- здатність аналізувати структуру даних на різних рівнях абстракції;

- легкість отримання результату у вигляді дендрограми;
- не потребує обов'язково вказувати кількість маркерів;

Недоліки ієрархічної кластеризації:

- має високі обчислювальні витрати;
- досить обмежене використання, тобто не можна змінювати кластеризацію після об'єднання або розділення маркерів.

Зазвичай така кластеризація використовується для виділення різних шарів тканин або клітин для медичних знімках. Або наприклад для аналізу геологічних зображень для виявлення природних форм. Також використовується для аналізу соціальних мереж у виявленні груп людей зі схожими характеристиками.

Спектральна кластеризація побудована на основі даних, використовуючи спектральні властивості матриці суміжності графа.

Процес роботи спектральної кластеризації:

1. Створюється граф, де кожен піксель даних є вузлом, а ваги ребер визначаються подібністю між пікселями.
2. Далі обчислюється власні значення та власні вектори матриці Лапласа цього графа.
3. На основі власних векторів даних виконується кластеризація.

Переваги спектральної кластеризації:

- достатньо висока точність у класифікації складних структур даних;
- здатність виявляти маркери з нерегулярними формами.

Недоліки спектральної кластеризації:

- високі обчислювальні витрати;
- потреба у попередньому визначенні кількості маркерів;
- залежність від вибору метрики подібності.

Один з методів, який побудований на основі густини це Mean Shift [24], простими словами, він намагається знайти густі області в даних, які можуть відповідати маркерам.

Процес роботи методу Mean Shift:

1. Спочатку обирається вікно з певним радіусом, яке буде змінювати положення по простору даних.
2. Далі це вікно переміщується по області, де є найбільше густини даних.
3. І такий процес повторюється до тих пір, доки обране вікно не зупиниться на найбільшому набору пікселів.

Переваги Mean Shift:

- здатність виявляти маркери будь-якої геометричної фігури без потреби у попередньому визначенні кількості маркерів;
- висока точність у центруванні маркерів;
- легкий у налаштуванні та використанні.

Недоліки Mean Shift:

- високі обчислювальні витрати;
- потреба у параметрі вибору вікна;
- не ефективний при даних з різним рівнем щільності.

[25], ще один з методів кластеризації, який побудований на основі графів, проте цей метод використовує графи для представлення структури даних та знаходить оптимальні розрізи, які мінімізують загальну вагу зав'язків між різними маркерами.

Процес роботи методу кластеризації Normalized Cuts:

1. Спочатку створюється граф, у якому кожен піксель даних є вузлом, а ваги ребер визначаються подібністю між маркерами.
2. Далі створюється матриця Лапласа графа.
3. І на останок використовується алгоритм оптимізації для знаходження розрізів.

Переваги Normalized Cuts:

- висока точність у сегментації;
- здатність зберігати цілісність між об'єктами;
- ефективність у виявленні маркерів з нерегулярними формами.

Недоліки Normalized Cuts:

- високі обчислювальні витрати;

- потреба у виборі параметра, тобто не є універсальним;
- не ефективність при виборі метрики подібності.

1.3 Аналіз існуючих підходів до підвищення точності вимірювань у технічних системах

При аналізі існуючих підходів до підвищення точності вимірювань у технічних системах [26], основними напрямками дослідження є розвиток сенсорних технологій [27], методи калібрування та компенсації похибок, алгоритми обробки даних, системи управління якістю вимірювань, інтеграція різних методів та автоматизація вимірювань.

Сенсори - це пристрої, які збирають дані про об'єкти, наприклад камери, які роблять фотографії та відео. Так от, сучасні сенсори стали дуже точними і дуже детально передають інформацію. Відповідно, зображення та відео які отримуються є дуже чіткими, і чим чіткіше відео, тим чіткіше є визначення об'єктів.

Переваги:

- більш детальні та чіткі зображення;
- зменшення шуму в даних.

Недоліки:

- високі витрати на сучасні сенсори;
- проблеми з точністю при низькому освітленні та при швидкому русі об'єктів.

Навіть точніші сенсори можуть давати помилки. Калібрування - це процес налаштування сенсорів [28], щоб вони давали правильні результати. Компенсація похибок - це спроби виправлення цих помилок автоматично.

Наприклад якщо камера трохи зміщена або світло змінюється, це впливає на результат, впливає на точність. Калібрування допомагає виправити ці помилки, а компенсація похибок використовує програми для автоматичного коригування даних, враховуючи зміни умов.

Переваги:

- підвищення точність вимірювань;
- автоматичне виправлення помилок.

Недоліки:

- потреба у регулярному калібруванні сенсорів;
- високі обчислювальні витрати.

Алгоритм обробки даних - це програми, які аналізують зібрані дані і виділяють з них корисну інформацію. Використання сучасних алгоритмів, таких як машинне навчання та нейронні мережі.

Наприклад, згорткові нейронні мережі можуть навчитися розпізнавати об'єкти на зображеннях та відстежувати їхній рух з високою точністю. Фільтри Калмана [29] допомагають прогнозувати та коригувати позицію об'єктів у реальному часі, навіть якщо дані містять шум.

Переваги:

- висока точність у розпізнаванні та відстеженні об'єктів;
- можливість автоматизації складних процесів аналізу даних.

Недоліки:

- потреба у великих обсягах даних для навчання алгоритмів;
- високі обчислювальні витрати.

Ще одним досить популярним та цікавим способом є поєднання різних методів та технологій [30]. Наприклад поєднати оптичні сенсори разом з методами машинного навчання, і тоді можна отримати більш точні та надійні дані про рух об'єктів. Або наприклад використати багатовимірні підходи, де вимірювання відбувається у декількох напрямках та в результаті отримати більш чіткий результат про стан руху об'єкта.

Переваги:

- висока точність та здатність адаптуватися під різні задачі.

Недоліки:

- велика складність через інтеграцію різних систем;
- високі обчислювальні витрати.

1.4 Висновки по розділу 1 та постановка задачі

Було проведено огляд методів аналізу зображень, огляд методів кластеризації та аналіз існуючих підходів до підвищення точності вимірювань у технічних системах.

Проведений аналіз свідчить про широку розповсюдженість методів кластеризації та питання визначення об'єктів та виявлення зміщення між ними.

В результаті аналізу було прийнято рішення використовувати метод кластеризації k-середніх, оскільки він має безліч переваг, серед яких простота, висока точність, висока швидкість, універсальність в інтеграції з різними алгоритмами

Метою роботи є підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації.

Для досягнення мети було поставлено наступний перелік задач:

- провести аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектувати метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначити набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконати програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначити точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.

РОЗДІЛ 2 Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

2.1 Загальна структура методу кластеризації для аналізу зображень

Для вирішення задачі методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації використовується метод кластеризації k-середніх. Він автоматизує сегментацію зображення та спрощує виявлення і відслідковування об'єктів. Його процес поділяється на три кроки:

1. Обробка зображення.
2. Виявлення об'єктів.
3. Обчислення зміщення.

Кожен з кроків поділяється на три складові й кожен з яких потрібен для вирішення задачі шляхом застосування методу кластеризації. Загальна структура методу представлено на рисунку 2.1

Напочатку процесу методу з використанням методів кластеризації полягає в отриманні зображення з різних джерел, таких як цифрові камери, сканери, відеокамери або інтернет. Від того, яке саме зображення буде отримано на вході, залежить усі наступні процеси. Наприклад, якщо зображення достатньо чітке, має високу роздільною здатність, в результаті це зображення буде якісно сегментовано. Наприклад, зображення, отримані за допомогою цифрових камер, часто мають високу деталізацію і можуть бути використані для аналізу об'єктів у реальному часі. Сканери, навпаки, забезпечують статичні зображення з високою точністю.

Способи отримання зображення продемонстровано на рисунку 2.2.

Крок 1: Обробка зображення (1.1 Фільтрація шумів).

Наступним кроком є обробка зображення. Цей крок відповідає за підготовку зображення для подальшого аналізу шляхом усунення шумів, покращення контрастності та нормалізації зображення. Обробка зображень робить об'єкти на зображенні чіткішими та дозволяє краще їх розпізнати в наступних кроках.

Обробка зображення має кілька своїх кроків, які продемонстровано на рисунку

2.3.

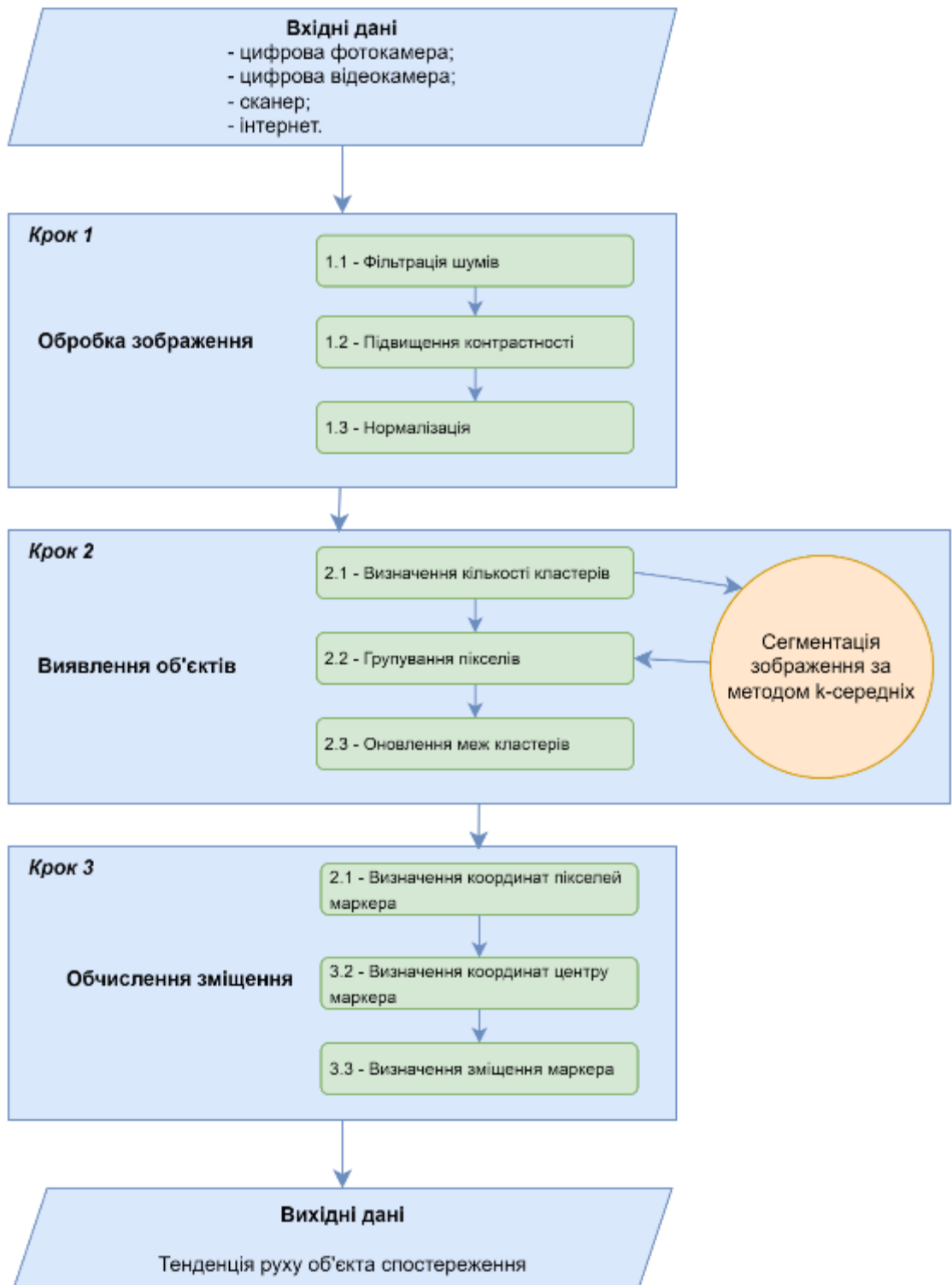


Рисунок 2.1 - Загальна схема кроків визначення зміщення об'єктів з використанням методу кластеризації



Рисунок 2.2 - Способи отримання зображень для аналізу

Крок 2: Виявлення об'єктів.

Далі використовується кластеризація k-середніх для поділу зображення на окремі маркери, кожен з яких відповідає окремим об'єктам або їх частинам. Кластеризація k-середніх автоматично групує пікселі на основі їхньої інтенсивності або кольорових характеристик. Після сегментації зображення кожен маркер вважається окремим об'єктом. На цьому кроці визначаються геометричні центри об'єктів, їхні контури, а також їхнє початкове положення на зображенні. Це необхідні дані для подальшого відстеження зміщення кожного об'єкта між послідовними зображеннями.

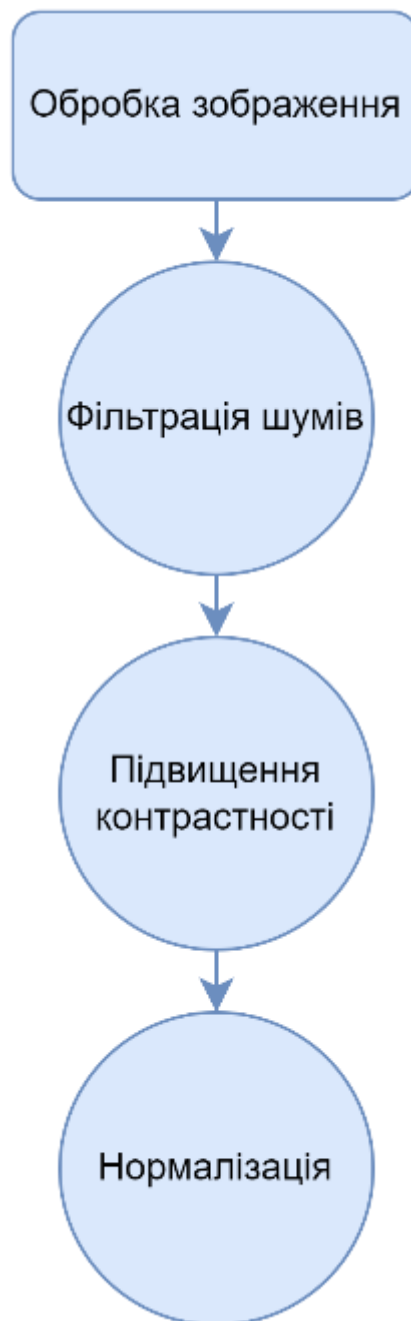


Рисунок 2.3 - Кроки попередньої обробки зображення

Виявлення об'єктів включає кілька кроків, зокрема пошук контурів об'єктів, побудову обмежуючих рамок навколо кожного виявленого об'єкта. У цьому кроці відбувається забезпечення структурованої інформації для аналізу їхнього зміщення. Від того, наскільки точно буде визначено центри об'єктів та їхні контури, настільки точні будуть подальші розрахунки зміщення.

Виявлення об'єктів має кілька своїх кроків, які продемонстровано на рис 2.4.



Рисунок 2.4 - Кроки виявлення об'єктів

Крок 3: Обчислення зміщення

На завершення процесу кластеризації відбувається визначення зміщення об'єктів між кількома зображеннями. Використовуючи координати центрів об'єктів, визначені на попередньому кроці, система обчислює зміщення шляхом порівняння їхніх положень на послідовних зображеннях. Після чого визначається величина та напрямок зміщення кожного об'єкта.

Процес обчислення зміщення включає визначення центрів кластерів на другому зображенні за допомогою алгоритму k-середніх, порівняння їх з центрами кластерів на першому зображенні та обчислення векторів зміщення.

Цей крок має кілька своїх кроків, які продемонстровано на рисунку 2.5



Рисунок 2.5 - Обчислення зміщення об'єктів між кількома зображеннями

Після обчислення зміщення об'єктів система генерує вихідні дані, які включають інформацію про тенденцію руху кожного об'єкта.

2.2 Адаптація методу k-середніх

Метод k-середніх є достатньо ефективний, в тому числі математичною оптимізацією, та в тому, як він побудований. Простіше кажучи, математична частина методу мінімізує внутрішню дисперсію пікселів навколо центрів маркерів. Через що, метод має високу точність у виявленні зміщення об'єктів, навіть тоді, коли використовується у задачах, де важлива точність і кожна похибка є серйозною.

Математичне формулювання методу k-середніх полягає в мінімізації функції втрат J , що описує суму квадратів відстаней між пікселями і центрами їхніх маркерів. Функція втрат визначається наступним чином:

$$J = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2, \quad (2.1)$$

де C_j позначає множину пікселів, що належать до маркера j , а $\|x_i - \mu_j\|^2$ - квадрат евклідової відстані між пікселем x_i та центром маркера μ_j . Така постановка задачі досягає глобальної оптимізації положень центрів маркерів.

Ітераційний процес оптимізації у методі k-середніх складається з трьох основних кроків: початкова ініціалізація центрів маркерів, розподіл пікселів за маркерами відповідно до найменшої відстані до центрів, і подальше оновлення центрів як середніх значень пікселів у кожному маркері. Процес триває до досягнення збіжності, тобто коли зміни в положеннях центрів між ітераціями стають незначними.

Нехай μ_j^t позначає положення центру маркера j на кадрі t , а μ_j^{t+1} - на наступному кадрі $t + 1$. Величина зміщення центру маркера між двома кадрами обчислюється як:

$$\Delta d_j = \|\mu_j^{t+1} - \mu_j^t\|, \quad (2.2)$$

де Δd_j є евклідовою відстанню між положеннями центру маркера на двох кадрах.

Однією з особливостей методу є використання передових методів обробки зображень перед кластеризацією. Фільтрація шумів знижає випадкові спотворення, які призводять до неточності у кластеризації. Підвищення контрастності та нормалізація зображень дають більш точне визначенню меж об'єктів. Це впливає на точність кластеризації та, відповідно, на точність визначення зміщення об'єктів на зображеннях. Для оптимальної кількості маркерів у задачах кластеризації застосовується метод ліктя. Цей метод ґрунтується на побудові графіка залежності функції втрат J від кількості маркерів k . Збільшення кількості маркерів зменшує значення J , проте після певного пікселя зменшення стає менш значимим. Піксель, де спостерігається різка зміна градієнта, і є оптимальною для вибору кількості маркерів.

2.3 Кроки реалізації методу з використанням кластеризації

Процес реалізації поділяється на кілька кроків, кожен з яких відповідає певному кроку обробки зображень і аналізу результатів. Метод кластеризації, зокрема k -середніх, групує пікселі на основі їхніх характеристик для визначення об'єктів на зображенні та подальшого аналізу їхнього зміщення (рис. 2.6).

Крок 1. Обробка зображень.

Напочатку йде підготовка даних, яка включає отримання вхідних зображень, обробку та нормалізацію. Це необхідно для забезпечення коректності подальшого аналізу. Зображення можуть бути отримані різними шляхами (цифрова камера, сканер, інтернет тощо), після чого вони обробляються для подальшого використання у процесі кластеризації.

Крок 2. Виявлення об'єктів.

Далі визначаються кількість маркерів, скільки необхідно провести ініціалізацію маркерів. Для цього використовується метод k -середніх який визначає оптимальні стартові маркери. На цьому кроці кожен маркер призначається випадковим чином (рис. 2.6).

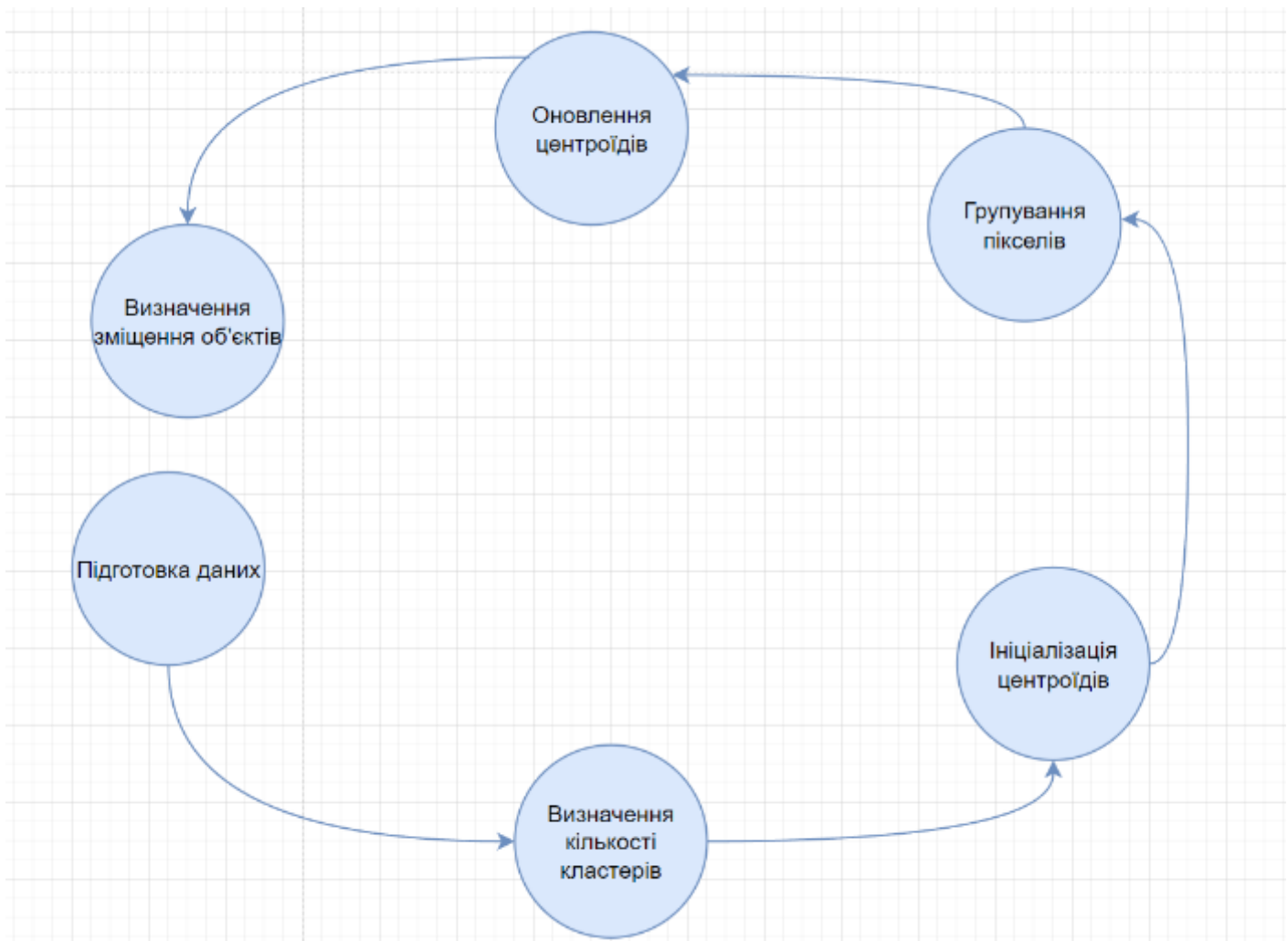


Рисунок 2.6 - Загальний процес реалізації кластеризації для вирішення задачі зміщення об'єктів

Наступним кроком є групування пікселів на основі відстаней до найближчих маркерів. Цей процес здійснюється за допомогою евклідової відстані між пікселем і маркером (рис. 2.8). Пікселі, що належать до одного маркера, будуть групуватись разом, утворюючи сегмент об'єкта на зображенні.

Після первинного групування пікселів до кластерів необхідно оновити позиції маркерів, щоб відобразити нові середні координати пікселів, що належать до кожного кластеру. Це здійснюється шляхом обчислення середнього значення координат пікселів у кожному кластері, після чого маркер зміщується в нову позицію, де він найбільш точно відображає центр мас пікселів цього кластеру. (рис. 2.9).

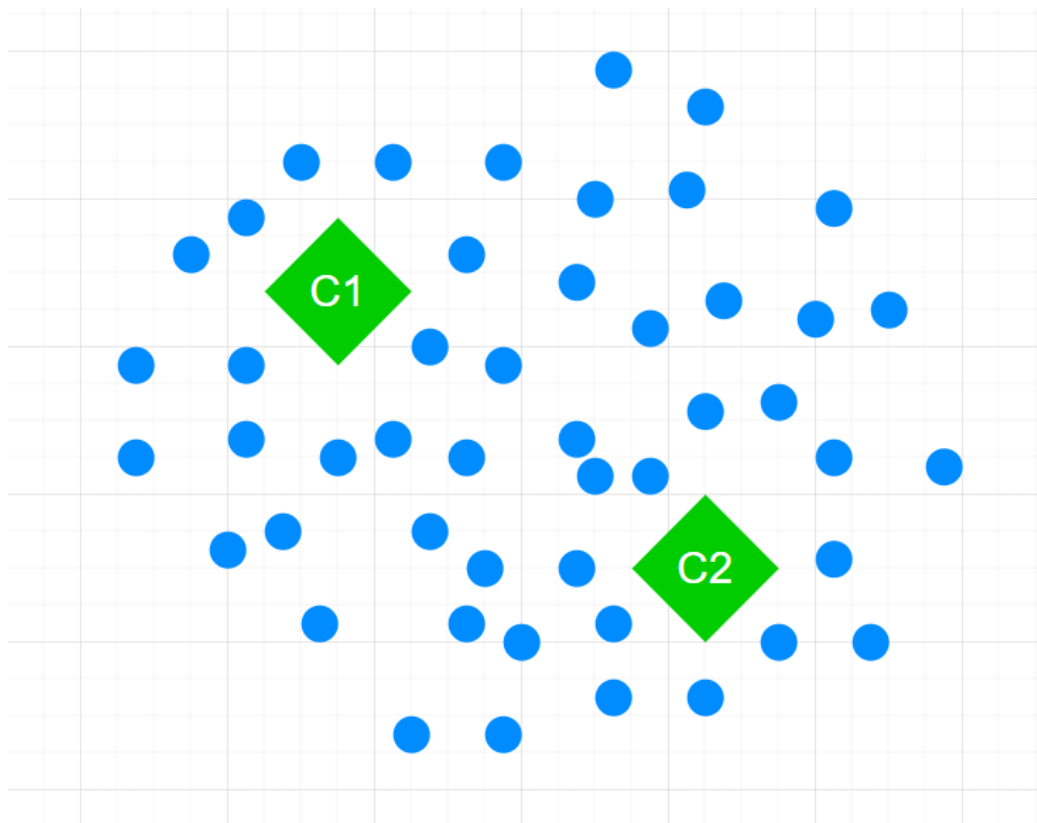


Рисунок 2.7 - Ініціалізація маркерів

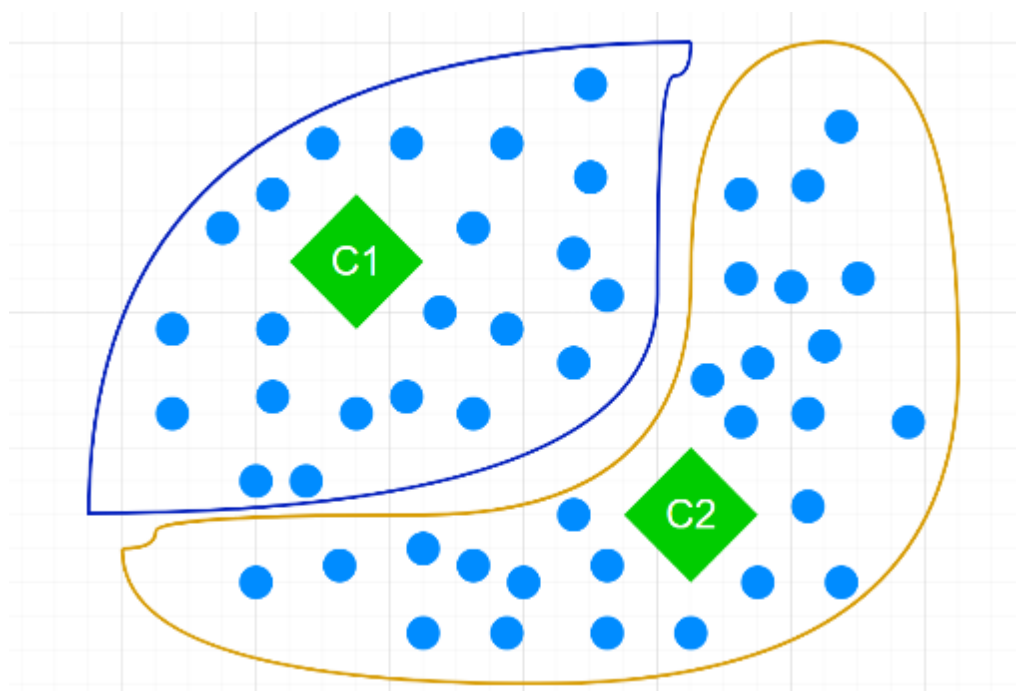


Рисунок 2.8 - Процес групування пікселів за найближчими маркерами зображення

Цей процес оновлення маркерів повторюється кілька разів до тих пір, доки зміни у положеннях маркерів між ітераціями не стають незначними, тобто досягнута збіжність алгоритму. Завдяки цьому маркери стабілізуються в оптимальних позиціях для забезпечення точного розділення об'єктів на зображенні та високої точності подальшого аналізу їхнього зміщення.

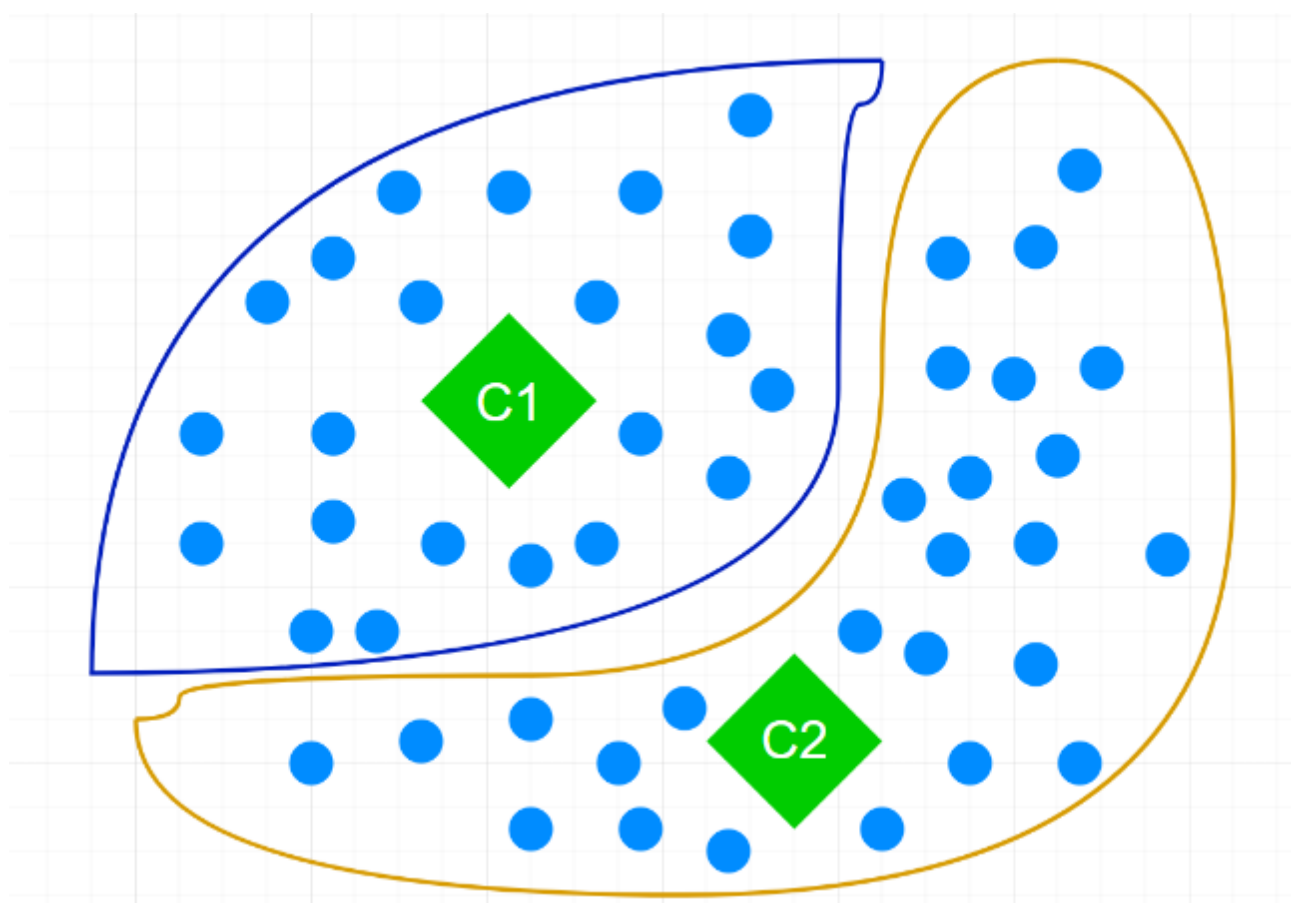


Рисунок 2.9 - Процес оновлення положень маркерів на основі середніх координат пікселів

Крок 3. Виявлення об'єктів

На останок відбувається визначення зміщення об'єктів між кількома послідовними зображеннями. Використовуючи координати центрів об'єктів, визначені на попередніх кроках, система обчислює зміщення шляхом порівняння положень об'єктів на різних кадрах.

В результаті визначається напрямок та величина зміщення кожного об'єкта.
(рис. 2.10).

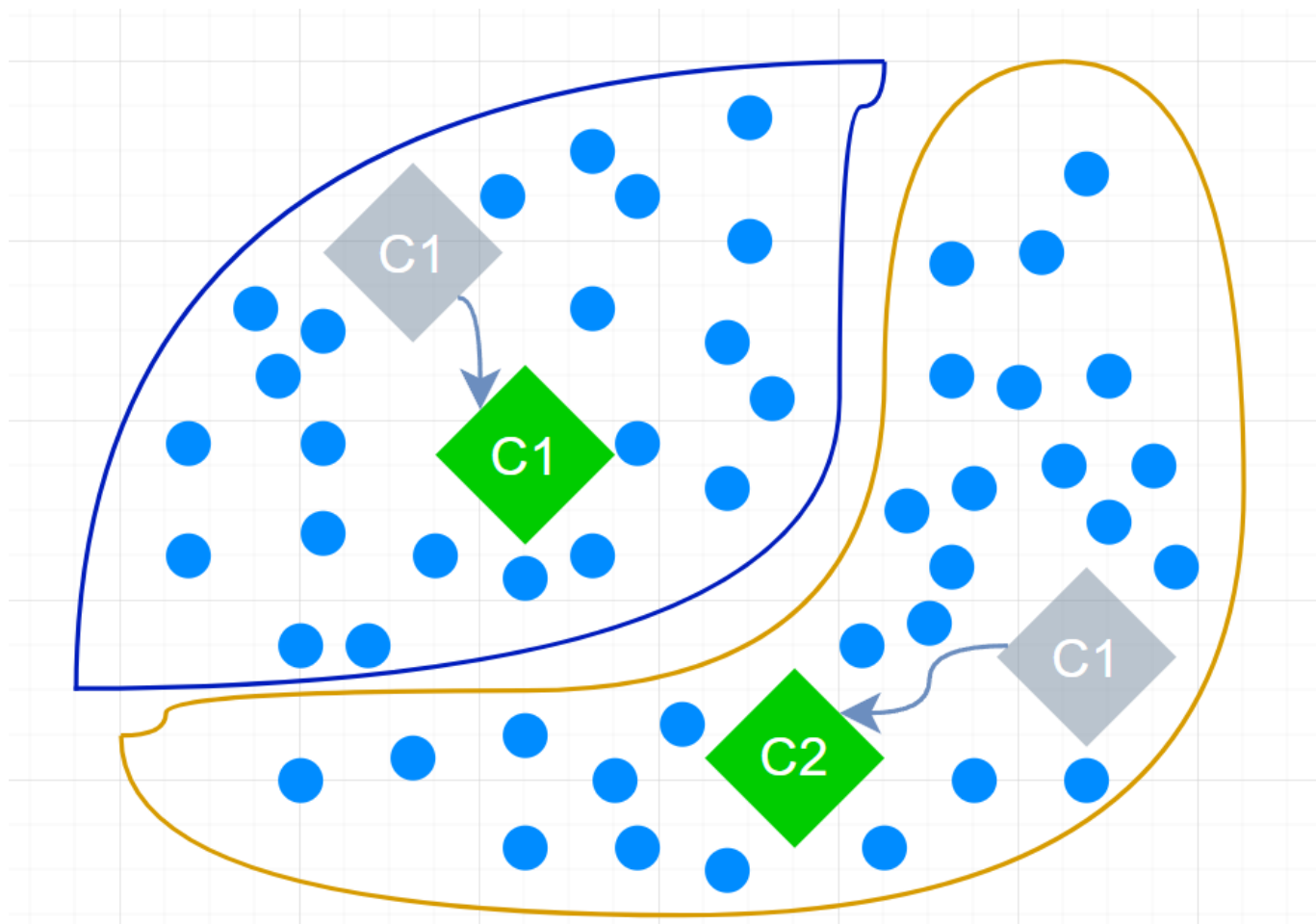


Рисунок 2.10 - Опис процесу визначення зміщення об'єктів на основі зміни положень маркерів між двома зображеннями

Процес визначення зміщення включає наступні кроки. Спочатку визначаються центри кластерів на першому зображенні, а потім аналогічним чином визначаються центри на другому зображенні.

Далі відбувається порівняння цих центрів, де кожен центр з першого зображення підставляється з найближчим центром з другого зображення. Після цього обчислюється евклідова відстань між відповідними центрами кластерів, що представляє вектор зміщення для кожного об'єкта.

2.4 Математичне забезпечення методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використання методів кластеризації

Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації спрямований на отримання кількісних значень зазначених переміщень. Кількісне значення для визначення кількісного значення прямолінійного руху повинно включати в основному геометричні розміри самого об'єкта та його складових.

На рисунках 2.11-2.13 продемонстровано геометричні фігури у вигляді квадрата, трикутника та кола в формі маркера. На рисунках продемонстровано те, як буде обчислюватися кожен піксель маркера відповідна стрілка зображена на кожному рисунку, та проведена жовта лінія для демонстрації контурів геометричних фігур.

На рисунку 2.11 представлено геометрична фігура квадрата у формі маркера.

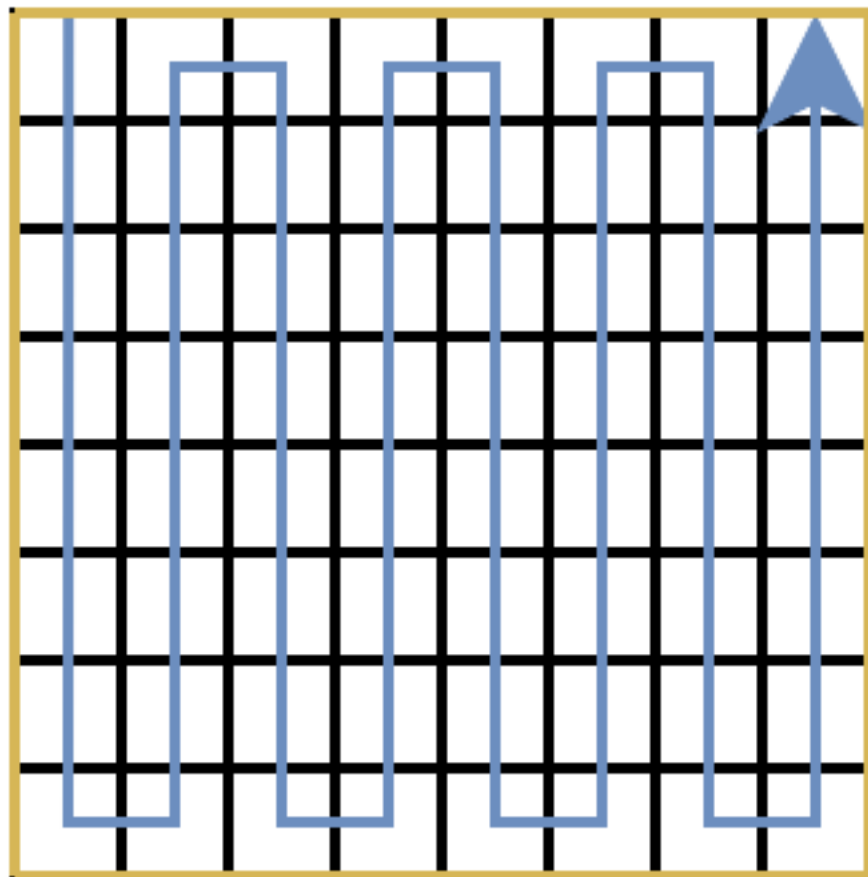


Рисунок 2.11 - Маркер у формі квадрата

Через що необхідно використовувати фотоматеріалів як початкові дані у зазначеній темі та використовувати цифрові засоби для формування зображень в комплексі з обчислювальною технікою. Після чого передати результати в програмне забезпечення.

Такий спосіб в рамках методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації дозволяє виконувати усі необхідні задачі, починаючи від фіксації об'єктів до їх фінального аналізу.

На рисунку 2.12 представлено геометрична фігура трикутника у формі маркера.

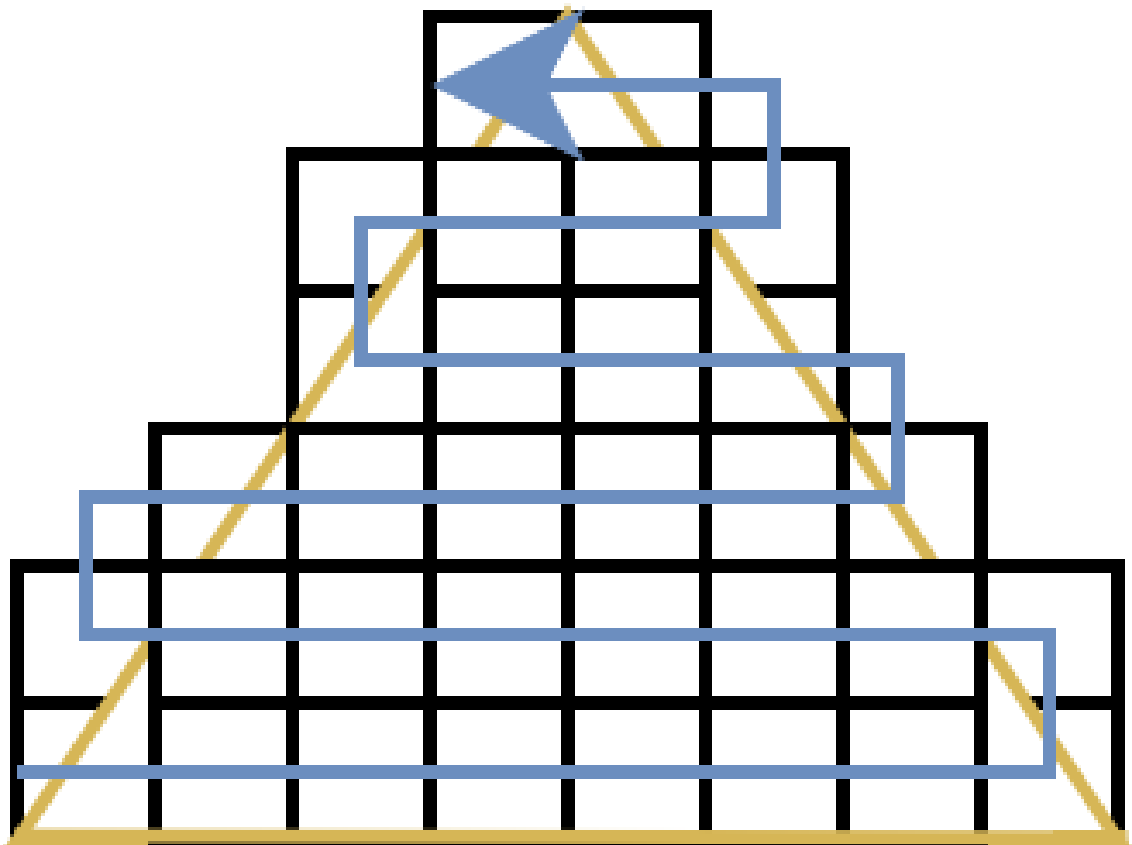


Рисунок 2.12 - Маркер у формі трикутника

На рисунку 2.13 представлено геометрична фігура кола у формі маркера.

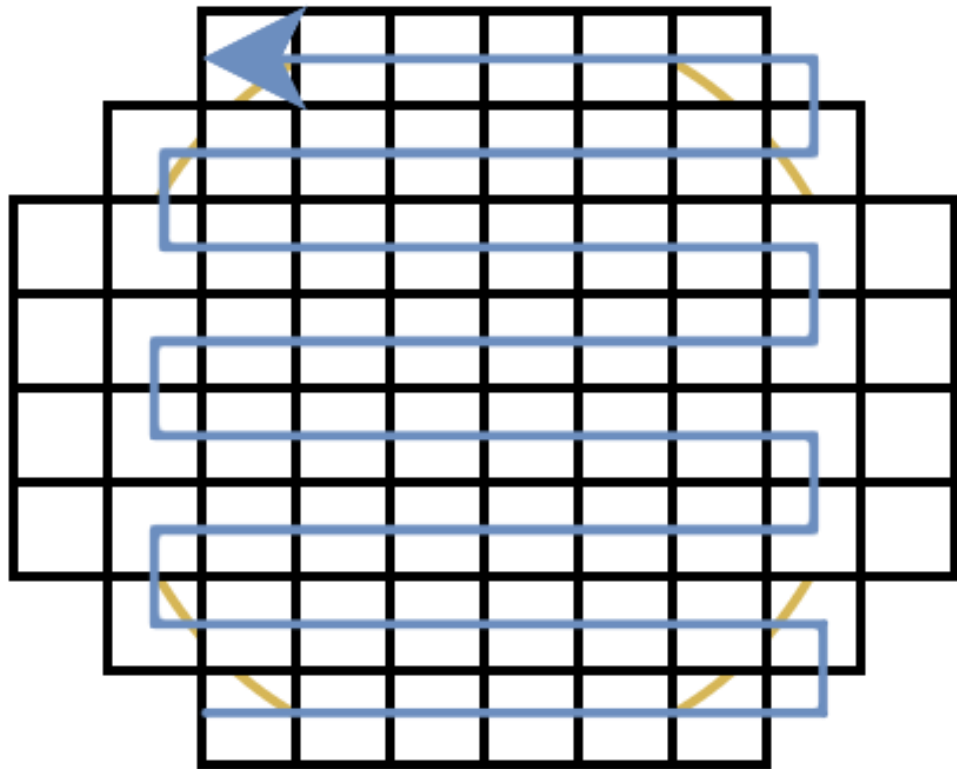


Рисунок 2.13 - Маркер у формі кола

Геометричний центр маркера визначається обчисленням середнього арифметичного координат для кожного пікселя на зображенні маркера. Це виникає при наближенні або віддаленні маркера. При наблизженні або віддаленні маркера виникає не точність оцінки паралельності між площиною руху маркера та площиною датчика зображення. (рис. 2.14).

Положення маркера визначається його геометричним центром, який розраховується на основі наступного співвідношення:

$$L = \sqrt{(X_{\text{цм}})^2 + (Y_{\text{цм}})^2}, \quad (2.3)$$

де $X_{\text{цм}}$, $Y_{\text{цм}}$ - координати геометричного центру маркера.

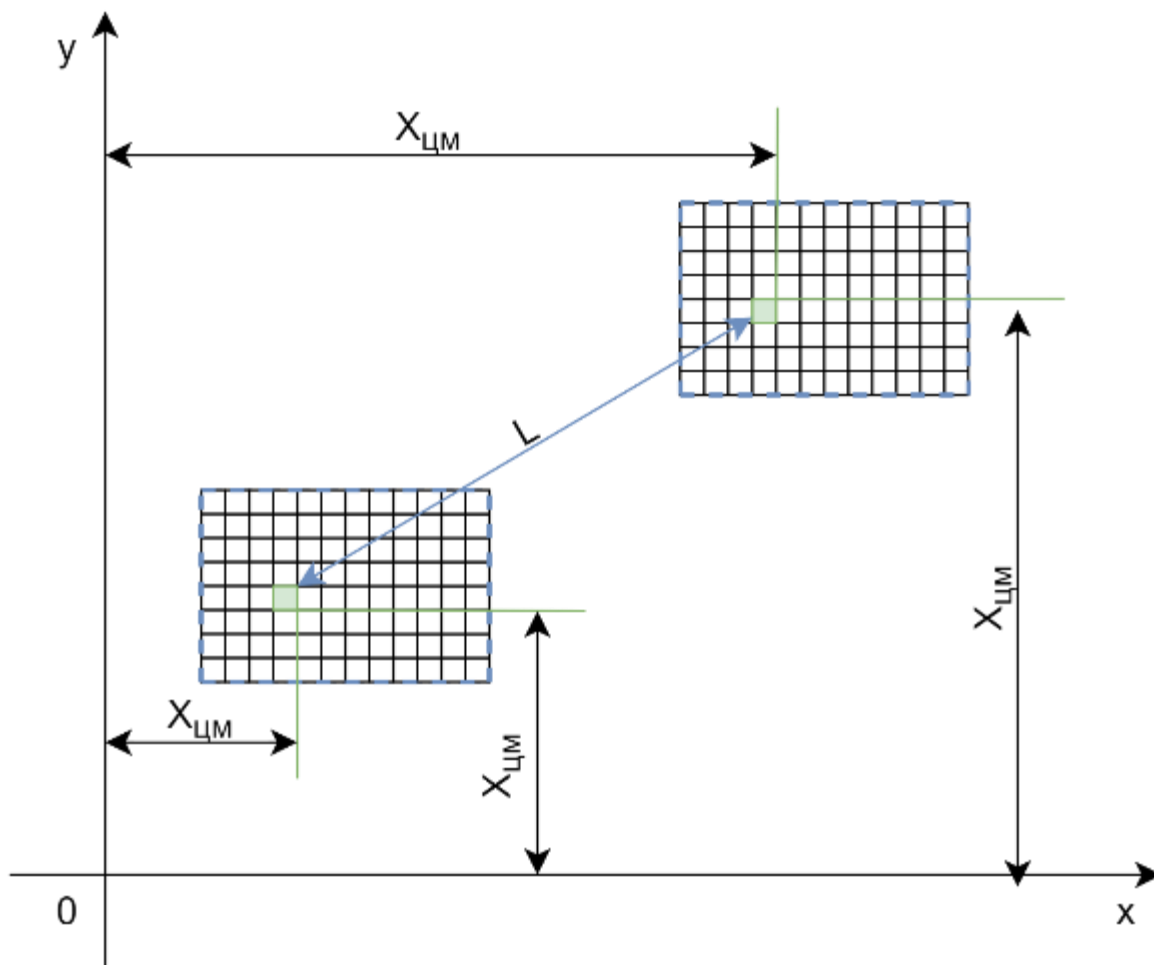


Рисунок 2.14 - Загальна схема вимірювань

На рисунку 2.14 продемонстровано розрахунок геометричного центру маркера.

2.5 Критерії покращення визначення зміщення

Метою розробки методу є підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. Задля чисельної оцінки покращення точності можуть бути використані такі статистичні метрики:

- експериментальна оцінка математичного сподівання зміщення:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i, \quad (2.4)$$

де N - кількість експериментів вимірювань, t_i - значення зміщення у i -му експериментів.

– експериментальна оцінка дисперсії зміщення:

$$s_t^2 = \frac{1}{N-1} \sum_{i=1}^N (t_i - \bar{t})^2, \quad (2.5)$$

де N - кількість експериментів або вимірювань, t_i - значення зміщення у i -му експериментів t_i - експериментальна оцінка математичного сподівання зміщення.

– експериментальна оцінка середнього квадратичного відхилення:

$$S_t = +\sqrt{s_t^2}, \quad (2.6)$$

де s_t^2 - дисперсія зміщення

– експериментальна оцінка коефіцієнту варіації:

$$v_t = \frac{S_t}{\bar{t}}, \quad (2.7)$$

де S_t - середнє квадратичне відхилення зміщення, \bar{t} - середня значення зміщення.

– експериментальна оцінка відносної похибки:

$$\Delta_{\max} = \frac{|\bar{t}_\phi - \bar{t}_p|}{\bar{t}_p} * 100\%, \quad (2.8)$$

де \bar{t}_ϕ - фактичне середнє значення зміщення, \bar{t}_p – прогнозоване середнє значення зміщення.

Описані критерії покращення визначення зміщення будуть використовуватися у наступних розділах під час тестування програмної реалізації.

Висновки до розділу 2

Було розроблено метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням кластеризації. Підвищення точності визначення величини зміщення об'єкту на зображеннях досягається за рахунок використання методу кластеризації k-середніх при визначенні контурів маркеру, переміщення якого приймаються ідентичними переміщенням об'єкту спостереження.

Побудовано загальну структуру методу підвищення точності визначення зміщення об'єктів з адаптацією та імплементацією методу k-середніх з покроковою його деталізацією.

Визначено критерії оцінки підвищення точності вимірювання зміщення за які прийнято статистичні показники.

РОЗДІЛ 3 Програмна реалізація методу підвищення точності визначення зміщення об'єктів

3.1 Архітектура та модулі програмної реалізації

Архітектура програмної реалізації складається з восьми основних модулів, кожен з яких має свою задачу і виконує важливу роль у програмі.

Основна архітектура система зображено на рисунку 3.1.



Рисунок 3.1 - Загальна архітектура програмної реалізації

Усе починається з модуля завантаження та обробки зображення. Цей модуль відповідає за завантаження зображень з різних пристроїв та джерел, це може бути фотокамери, сканери, відеокамери, принтери або інші пристрої, які передають зображення. Програма працює таким чином, що користувач може самостійно обрати зображення в файловій системі, після чого починається маніпуляція з зображенням.

Власне, весь процес модуля продемонстровано на рисунку 3.2.

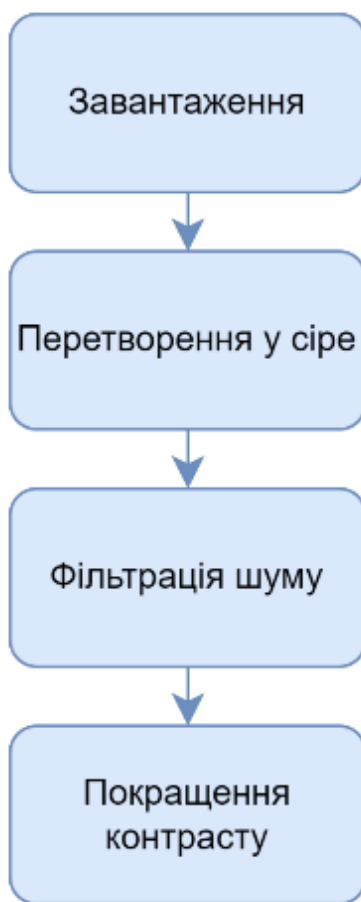


Рисунок 3.2 - Процес попередньої обробки зображень

Після завершення роботи модуля обробки зображення, починає працювати наступний модуль, модуль кластеризації пікселів. Спочатку, необхідно отримати зображення, які передав перший модуль. Далі йде групування пікселів за допомогою алгоритму k-середніх. Створюються маркери, які представляють собою різні класи пікселів.

Процес цього модуля продемонстровано на рисунку 3.3.

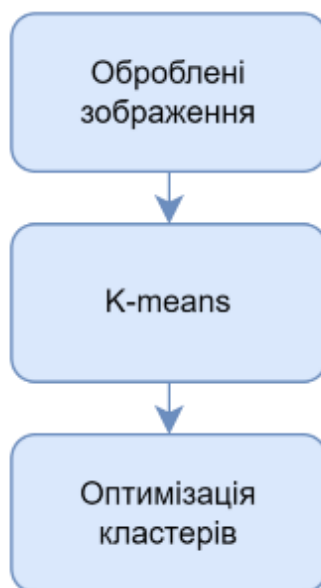


Рисунок 3.3 - Модуль кластеризації пікселів

Наступний модуль відповідає за виявлення об'єктів і побудову контурів. Відповідно, спочатку необхідно отримати результати кластеризації від минулого модуля, а далі йде пошук контурів, адже тут необхідно дуже точно визначити межі об'єктів на зображенні. Після чого на основі цих меж побудувати рамки навколо об'єкта. Процес роботи модуля продемонстровано на рисунку 3.4.



Рисунок 3.4 - Виявлення об'єктів за результатами кластеризації

Наступний модуль відповідає за аналіз зміщення. Спочатку, цей модуль отримує результати визначених об'єктів на двох зображення. І далі на основі отриманих даних, модуль обчислює вектори зміщення для кожного об'єкта, якщо однакові об'єкти були знайдені. Після чого визначається напрямок і величину зміщення.

Процес роботи модуль продемонстровано на рисунку 3.5.



Рисунок 3.5 - Процес аналізу та відстеження зміщення об'єктів

Далі йде модуль звітності. Цей модуль відповідає за надання якісного результату користувачеві. Тобто, модуль отримує готові дані від інших модулів та формує зрозумілу інформацію для користувача. Сюди входить звітність про номер об'єкта, геометрична фігура об'єкта, зміщення в пікселях, зміщення в міліметрах, графічне сегментоване зображення, два початкових зображення з намальованими контурами навколо визначених об'єктів, а також зберігання результатів роботи програми в окремій папці.

Процес роботи модуля продемонстровано на рисунку 3.6.



Рисунок 3.6 - Візуалізація результатів аналізу зміщення об'єктів

Наступний модуль відповідає за те, як працює як інтерфейс, який бачить перед собою користувач, після того, як програма запустилась. На цьому інтерфейсі користувач має можливість завантажити зображення, обрати кількість кластерів інші додаткові дії.

Процес роботи модуля продемонстровано на рисунку 3.7.

Ще одним з наступних модулів, це модуль моніторингу, він відповідає за збір інформації, за те, щоб збирати якомога більше інформації про кожен крок програми, про кожну помилку, про різний стан програми, який може змінюватися постійно. Завдяки цьому модулю, можна легко працювати з помилками, які виникають під час роботи з програмою. Модуль моніторингу також відповідає за вивід помилок, наприклад якщо обрано лише одне зображення, але користувач намагається запустити програму. Цей момент фіксується і повідомляється користувачеві про причину помилки.

Процес роботи модуля продемонстровано на рисунку 3.8.

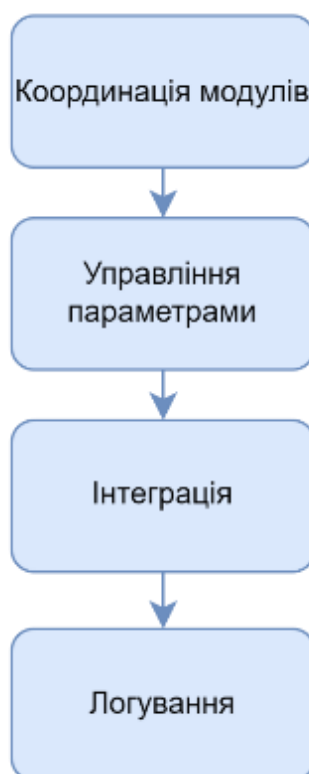


Рисунок 3.7 - Модуль управління та інтеграції



Рисунок 3.8 - Модуль моніторингу

3.2 Особливості програмної реалізації методу з кластеризацією

Програмна реалізація методу підвищення точності визначення зміщення об'єктів була розроблена на мові програмування Python [33] та за допомогою таких бібліотек як OpenCV [34], scikit-learn [35], Tkinter [36], PIL [37], skimage [38] та інших.

Відповідно, спочатку програма отримує зображення, після чого воно конвертується з кольорового BGR у Lab [39]. Для роботи з шумом використовується метод Гауса, після чого виконується нормалізація яскравості для підвищення контрастності.

Для того, щоб витягнути текстурні ознаки з зображення, використовується метод LBP [40], він допомагає виділяти структурні характеристики об'єктів на зображенні. А вже для кластеризації зображення використовується метод кластеризації k-середніх, відповідно до його роботи, він розділяє зображення на оптимальну кількість маркерів.

Після процесу кластеризації кожен маркер має свою площу, параметри кутів, співвідношення сторін та кількість вершин для визначення геометричної фігури. А для того, аби зіставити об'єкти між двома зображеннями, використовується алгоритм Гунського, для роботи він використовує центри маркерів між об'єктами.

Далі, коли вже відоме зміщення всіх об'єктів, важливо правильно перевести це зміщення з пікселів у міліметри. Для цього використовується або відомий масштаб, або цей масштаб визначається за допомогою алгоритму калібрування.

Після чого відбувається візуалізація результатів аналізу у вигляді обведення контурів об'єкта, квадрата, трикутника або кола. Відповідні результати зберігаються у папці під назвою result, де можна ознайомитися з результатами.

Графічний інтерфейс побудовано на основі бібліотеки Tkinter. Реалізований інтерфейс продемонстрований на рисунку 3.9. Як видно на рисунку, інтерфейс дозволяє завантажувати зображення, налаштовувати параметри кластеризації, вводити відомі дані калібрування та переглядати результати роботи.

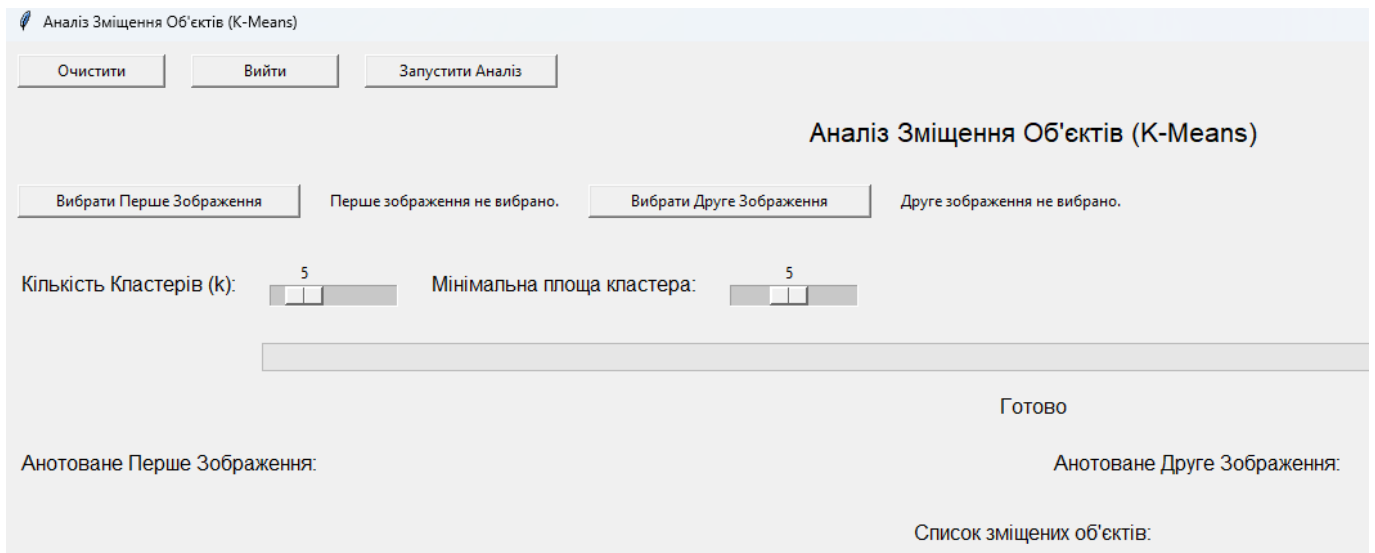


Рисунок 3.9 - Інтерфейс програмної реалізації

Для ефективної та плавної роботи процес роботи програми виконується в окремому потоці, це необхідно для того, аби користувач міг взаємодіяти з інтерфейсом навіть під час роботи програми.

3.3 Функціональні можливості системи для аналізу зміщення об'єктів

Програмна реалізація дозволяє користувачам завантажувати зображення різних форматів, до яких входить JPEG, PNG, BMP, TIFF через відповідні кнопки інтерфейсу. Після завантаження система автоматично проводить обробку зображень, зміну розміру, конвертацію кольорового простору, згладжування шумів та нормалізацію яскравості.

Користувач має можливість налаштувати кількість маркерів, мінімальну площу маркеру за допомогою спеціального функціоналу. Також у системі передбачено введення відомого масштабу зображення.

Після чого користувач може запустити аналіз зображень, паралельно виконуючі якісь дії у інтерфейсі, оскільки програма працює у фоновому потоці.

Після завершення аналізу результати відображаються у вигляді анотованих зображень. На кожному зображенні позначені знайдені зміщені об'єкти у вигляді

обводу контурів об'єкта. Крім цього, програма виводить результати аналізу у окреме вікно, у якому написано зміщення у пікселях, у міліметрах та назву об'єкта.

Результати роботи також зберігають у окремій папці result для того, аби результати можна було переглянути пізніше.

Інтерфейс також має функціональність очищення наявних даних, очищення результатів для того, аби легко керувати процесом аналізу.

3.4 Тестування програмної реалізації методу

Для забезпечення високої точності та надійності розробленої програмної реалізації було проведено комплексне тестування. Кожен тест-кейс розроблений з метою перевірки коректності роботи окремих модулів системи, а також їх взаємодії між собою. Для перевірки процесу завантаження зображень з різних джерел було створено тест-кейс (Таблиця 3.1), який забезпечує коректну роботу модуля завантаження та підтримку різних форматів файлів. Для перевірки процесу перетворення кольорових зображень у відтінки сірого було створено тест-кейс (Таблиця 3.2), який оцінює точність та якість цього процесу. Для перевірки ефективності фільтрації шуму на зображеннях було створено тест-кейс (Таблиця 3.3), який оцінює здатність зменшувати шум без втрати важливих деталей. Для оцінки ефективності процесів покращення контрастності та нормалізації освітлення було створено тест-кейс (Таблиця 3.4), який перевіряє рівномірність освітлення та підвищення контрастності зображення після обробки. Для оцінки ефективності кластеризації пікселів було створено тест-кейс (Таблиця 3.5), який перевіряє правильність сегментації зображення на об'єкти та фон. Для перевірки ініціалізації центрів кластерів було створено тест-кейс (Таблиця 3.6), який оцінює правильність вибору початкових точок та стабільність кластеризації при різних умовах. Для перевірки процесу виявлення контурів об'єктів було створено тест-кейс (Таблиця 3.7), який оцінює точність визначення меж об'єктів. Для перевірки точності обчислення геометричних центрів об'єктів було створено тест-кейс (Таблиця 3.8), який оцінює відповідність обчислених центрів фактичним позиціям об'єктів.

Для перевірки здатності системи точно відстежувати зміщення об'єктів між послідовними зображеннями було створено тест-кейс (Таблиця 3.9), який оцінює правильність обчислення векторів зміщення. Для перевірки коректності візуалізації векторів зміщення було створено тест-кейс (Таблиця 3.10), який оцінює правильність відображення напрямку та величини зміщення об'єктів на зображеннях. Для перевірки функціональності генерації графічних звітів було створено тест-кейс (Таблиця 3.11), який оцінює правильність створення та збереження звітів з результатами аналізу. Для перевірки гнучкості системи у налаштуванні параметрів кластеризації було створено тест-кейс (Таблиця 12), який оцінює вплив зміни кількості кластерів на результати кластеризації. Для перевірки можливості інтеграції системи з зовнішніми базами даних було створено тест-кейс (Таблиця 3.13), який оцінює коректність імпорту даних та їх подальшу обробку. Для перевірки здатності системи ефективно обробляти зображення високої роздільної здатності було створено тест-кейс (Таблиця 3.14), який оцінює час обробки та точність результатів при роботі з великими файлами. Для перевірки роботи системи з зображеннями, де об'єкти мають низьку освітленість, було створено тест-кейс (Таблиця 3.15), який оцінює ефективність попередньої обробки та точність визначення об'єктів у таких умовах. Для перевірки здатності системи одночасно відстежувати зміщення кількох об'єктів було створено тест-кейс (Таблиця 3.16), який оцінює точність визначення векторів зміщення для кожного об'єкта при їхньому одночасному переміщенні в різні напрямки.

Таблиця 3.1 - Тест-кейс ТК001

Тест-кейс ID: ТК001	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка завантаження зображень з різних джерел		
Вхідні дані: Зображення у форматах JPEG, PNG, BMP з цифрової камери, сканера та локальної файлової системи		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Відкрити модуль завантаження зображень. 2. Завантажити зображення з цифрової камери. 3. Завантажити зображення зі сканера. 4. Завантажити зображення з локальної файлової системи. 5. Перевірити відображення завантажених зображень у системі. 		Усі зображення успішно завантажуються без помилок та відображаються правильно у системі.
Результат виконання тест-кейсу: Пройдено успішно. Усі зображення були завантажені та відображені без помилок.		

Таблиця 3.2 - Тест-кейс ТК002

Тест-кейс ID: ТК002	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка перетворення кольорових зображень у відтінки сірого		
Вхідні дані: Кольорові зображення різних типів та розмірів		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити кольорове зображення у систему. 2. Запустити процес перетворення у відтінки сірого. 3. Перевірити результати перетворення. 		Кольорові зображення правильно перетворюються у відтінки сірого без втрати важливих деталей.
Результат виконання тест-кейсу: Пройдено успішно. Зображення були правильно перетворені у відтінки сірого, зберігаючи всі ключові деталі.		

Таблиця 3.3 - Тест-кейс ТК003

Тест-кейс ID: ТК003	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка фільтрації шуму		
Вхідні дані: Зображення з різними рівнями шуму		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити зображення з високим рівнем шуму. 2. Запустити фільтрацію шуму. 3. Оцінити якість зображення після фільтрації. 		Рівень шуму значно зменшується, зображення стає більш чітким без втрати важливих деталей.
Результат виконання тест-кейсу: Пройдено успішно. Рівень шуму був ефективно зменшений, а якість зображення покращена без втрати важливих деталей.		

Таблиця 3.4 - Тест-кейс ТК004

Тест-кейс ID: ТК004	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка покращення контрастності та нормалізації освітлення		
Вхідні дані: Зображення з нерівномірним освітленням		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити зображення з нерівномірним освітленням. 2. Запустити процес покращення контрастності та нормалізації освітлення. 3. Оцінити рівномірність освітлення та контрастність після обробки. 		Зображення має рівномірне освітлення та покращену контрастність.
Результат виконання тест-кейсу: Пройдено успішно. Освітлення стало більш рівномірним, а контрастність покращено.		

Таблиця 3.5 - Тест-кейс ТК005

Тест-кейс ID: ТК005	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка кластеризації пікселів		
Вхідні дані: Оброблені зображення з виділеними об'єктами та фоном		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Запустити процес кластеризації пікселів. Визначити оптимальну кількість кластерів. Оцінити якість сегментації об'єктів та фону. 		Пікселі правильно обробляються, об'єкти та фон виділяються чітко без помилок сегментації.
Результат виконання тест-кейсу: Пройдено успішно. Кластеризація була виконана точно, з чітким виділенням об'єктів та фону.		

Таблиця 3.6 - Тест-кейс ТК006

Тест-кейс ID: ТК006	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка ініціалізації центрів кластерів		
Вхідні дані: Зображення з різноманітною структурою об'єктів		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Запустити модуль кластеризації. Перевірити вибір початкових центрів кластерів. Оцінити стабільність та точність кластеризації при різних початкових умовах. 		Початкові центри кластерів вибираються оптимально, що забезпечує стабільність та високу точність кластеризації навіть при різних початкових умовах.
Результат виконання тест-кейсу: Пройдено успішно. Ініціалізація центрів кластерів була виконана оптимально, забезпечуючи стабільність та точність кластеризації.		

Таблиця 3.7 - Тест-кейс ТК007

Тест-кейс ID: ТК007	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка виявлення контурів об'єктів		
Вхідні дані: Оброблені зображення з чіткими об'єктами		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Запустити модуль виявлення об'єктів. Використати алгоритм пошуку контурів. Перевірити точність визначення контурів об'єктів. 		Контури об'єктів визначені точно, без пропусків чи зайвих артефактів, а обмежуючі рамки відповідають фактичним об'єктам.
Результат виконання тест-кейсу: Пройдено успішно. Контури були визначені точно, а обмежуючі рамки відповідали фактичним об'єктам.		

Таблиця 3.14 - Тест-кейс ТК014

Тест-кейс ID: ТК012	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка обробки великих зображень		
Вхідні дані: Зображення високої роздільної здатності (наприклад, 4000x3000 пікселів)		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Завантажити велике зображення у систему. Виконати попередню обробку зображення. Запустити процес кластеризації пікселів. Виконати виявлення об'єктів та аналіз зміщення. Перевірити час обробки та точність результатів. 		Система успішно обробляє велике зображення без значних затримок, забезпечуючи точність аналізу зміщення об'єктів.
Результат виконання тест-кейсу: Пройдено успішно. Велике зображення було оброблено ефективно, з часом обробки в межах допустимих значень та високою точністю аналізу.		

Таблиця 3.9 - Тест-кейс ТК009

Тест-кейс ID: ТК009	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка відстеження зміщення об'єктів між зображеннями		
Вхідні дані: Два послідовні зображення з об'єктами, що змістилися		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити перше та друге зображення у систему. 2. Виконати кластеризацію та виявлення об'єктів на обох зображеннях. 3. Запустити модуль аналізу та відстеження зміщення об'єктів. 4. Перевірити коректність обчислення векторів зміщення. 		Система точно визначає зміщення об'єктів, обчислюючи їхні вектори зміщення з правильною величиною та напрямком.
Результат виконання тест-кейсу: Пройдено успішно. Вектори зміщення були обчислені точно, відображаючи фактичні переміщення об'єктів.		

Таблиця 3.10 - Тест-кейс ТК010

Тест-кейс ID: ТК010	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка візуалізації векторів зміщення об'єктів		
Вхідні дані: Результати аналізу зміщення об'єктів		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Запустити модуль візуалізації та звітності. 2. Відобразити оброблені зображення з виділеними об'єктами. 3. Перевірити коректність відображення векторів зміщення. 		Вектори зміщення відображаються точно на зображеннях, вказуючи правильний напрямок та величину переміщення кожного об'єкта.
Результат виконання тест-кейсу: Пройдено успішно. Вектори зміщення були відображені точно, відповідаючи фактичним зміщенням об'єктів.		

Таблиця 3.11 - Тест-кейс ТК011

Тест-кейс ID: ТК011	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка генерації графічних звітів з результатами аналізу		
Вхідні дані: Результати аналізу зміщення об'єктів		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Запустити модуль візуалізації та звітності. Створити графічний звіт з результатами аналізу. Перевірити наявність всіх необхідних елементів у звіті (статистичні дані, графіки зміщення, траєкторії руху). Зберегти звіт у форматі PDF та CSV. 		Графічний звіт містить всі необхідні дані, оформлений чітко та зрозуміло, та успішно збережений у вибраних форматах.
Результат виконання тест-кейсу: Пройдено успішно. Звіт був створений з усіма необхідними елементами та успішно збережений у форматах PDF та CSV.		

Таблиця 3.12 - Тест-кейс ТК012

Тест-кейс ID: ТК012	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка налаштування параметрів кластеризації		
Вхідні дані: Зображення з різними характеристиками		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> Відкрити інтерфейс користувача. Змінити кількість кластерів (k) на різні значення. Запустити процес кластеризації з новими параметрами. Перевірити результати кластеризації відповідно до змінених параметрів. 		Зміна параметрів кластеризації впливає на результати кластеризації відповідно до очікувань.
Результат виконання тест-кейсу: Пройдено успішно. Зміна кількості кластерів відображалася у відповідних змінах результатів кластеризації, забезпечуючи необхідну гнучкість системи.		

Таблиця 3.13 - Тест-кейс ТК013

Тест-кейс ID: ТК013	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка інтеграції з зовнішньою базою даних		
Вхідні дані: Дані з зовнішньої бази даних щодо об'єктів		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Налаштувати з'єднання з зовнішньою базою даних. 2. Запустити імпорт даних з бази даних у систему. 3. Перевірити коректність імпортованих даних. 4. Виконати аналіз зміщення об'єктів на основі імпортованих даних. 		Дані успішно імпортовані з зовнішньої бази даних, система правильно їх обробляє та використовує у подальшому аналізі.
Результат виконання тест-кейсу: Пройдено успішно. Дані були імпортовані правильно, а система їх успішно обробила та використала у аналізі зміщення об'єктів.		

Таблиця 3.8 - Тест-кейс ТК008

Тест-кейс ID: ТК008	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка обчислення геометричних центрів об'єктів		
Вхідні дані: Виявлені об'єкти з чіткими контурами		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Запустити процес обчислення геометричних центрів об'єктів. 2. Перевірити правильність обчислених центрів у відповідності до фактичних положень об'єктів. 		Геометричні центри об'єктів обчислені точно, відповідно до їхніх фактичних положень на зображенні.
Результат виконання тест-кейсу: Пройдено успішно. Геометричні центри були обчислені точно, відповідно до фактичних положень об'єктів.		

Таблиця 3.15 - Тест-кейс ТК015

Тест-кейс ID: ТК015	Пріоритет: Середній	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка обробки зображень з низьким рівнем освітлення		
Вхідні дані: Зображення з недостатнім освітленням		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити зображення з низьким рівнем освітлення. 2. Виконати обробку зображення. 3. Запустити кластеризацію пікселів та виявлення об'єктів. 4. Перевірити якість сегментації. 		Система правильно обробляє зображення з низьким освітленням
Результат виконання тест-кейсу: Пройдено успішно. Об'єкти були виділені чітко, а їхні межі визначені точно навіть при низькому рівні освітлення.		

Таблиця 3.16 - Тест-кейс ТК016

Тест-кейс ID: ТК016	Пріоритет: Високий	Створено: 12.11.2024, Н. Вусатий
Назва: Перевірка відстеження зміщення кількох об'єктів одночасно		
Вхідні дані: Два послідовні зображення з кількома об'єктами, що рухаються в різні напрямки		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Завантажити перше та друге зображення з кількома рухомими об'єктами. 2. Виконати кластеризацію пікселів та виявлення об'єктів на обох зображеннях. 3. Запустити аналіз зміщення об'єктів. 4. Перевірити коректність визначення векторів зміщення для кожного об'єкта. 		Система точно визначає зміщення кожного об'єкта, незалежно від їхнього напрямку руху, без взаємного перекриття або помилок.
Результат виконання тест-кейсу: Пройдено успішно. Вектори зміщення кожного об'єкта були визначені точно, незалежно від напрямку їхнього руху.		

В результаті виконання тест-кейсів, усі пройдені успішно.

Висновки до розділу 3

Виконано програмну реалізацію методу підвищення точності визначення зміщення об'єктів за допомогою кластеризації.

Програмна реалізація виконана на основі модульній архітектурі системи, з 8 модулів, серед яких завантаження зображення, попередня обробка, кластеризація, виявлення об'єктів, аналіз зміщення, візуалізація, управління та логування.

Програмна реалізація розроблена на мові програмування Python та за допомогою таких бібліотек як OpenCV, scikit-learn, Tkinter, PIL, skimage та інших.

Описані функціональні можливості системи, включаючи завантаження зображень різних форматів, налаштування параметрів кластеризації, введення відомих зміщень для калібрування та перегляд анотованих зображень з результатами аналізу. Обробка зображень виконується у окремому потоці, а результати зберігаються в автоматично створеній папці для зручного доступу та подальшої обробки.

РОЗДІЛ 4 Дослідження методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

4.1 Оцінка точності визначення зміщення об'єктів

Для оцінки точності визначення зміщення об'єктів було зроблено 19 фотозображень з фотокамери.

Під час проведення фото, фотокамера знаходилась у фіксованому положенні та не змінювала своє положення під час проведення усіх фото. Геометричні фігури були розміщені на спеціальному фіксованому пристрою. Цей спеціальний пристрій дозволяє чітко змінювати положення об'єктів, що знаходяться на ньому.

Зображення були роздруковані на паперах різних контрастів. Перший на високому контрасті (білий колір), а інший на низькому контрасті (рожевий колір). На кожному контрасті зображено по три чорні геометричні фігури, які включають коло, квадрат та трикутник з чорним кольором.

На рисунку 4.1 продемонстровано приклад зробленого фото для експерименті.

Задля проведення тестів усі зображення були ретушовані. Під час цього процесу усі зображення зберегли початкові розміри та не були змінені за допомогою шумів, контрастності, світла або інших алгоритмів. Також, під цього процесу не було змінено положення усіх об'єктів на усіх зображеннях, на фото не було нічого зміщення і окрім ретушування жодних інших процесів не відбувалося.

Приклад ретушування зображень показано на рисунках 4.2-4.3. Для виконання ретушування використовувалась програма Paint. У програмі обирався спеціальний інструмент для визначення контрасту та кольору, після чого відбувалось ретушування зайвих геометричних фігур.

Усього було зроблено 19 зображень з інтервалом у 0.5 мм з точністю в 0.001 мм.

Під час виконання фото освітлення не було змінено та для всіх зображень є однаковим.

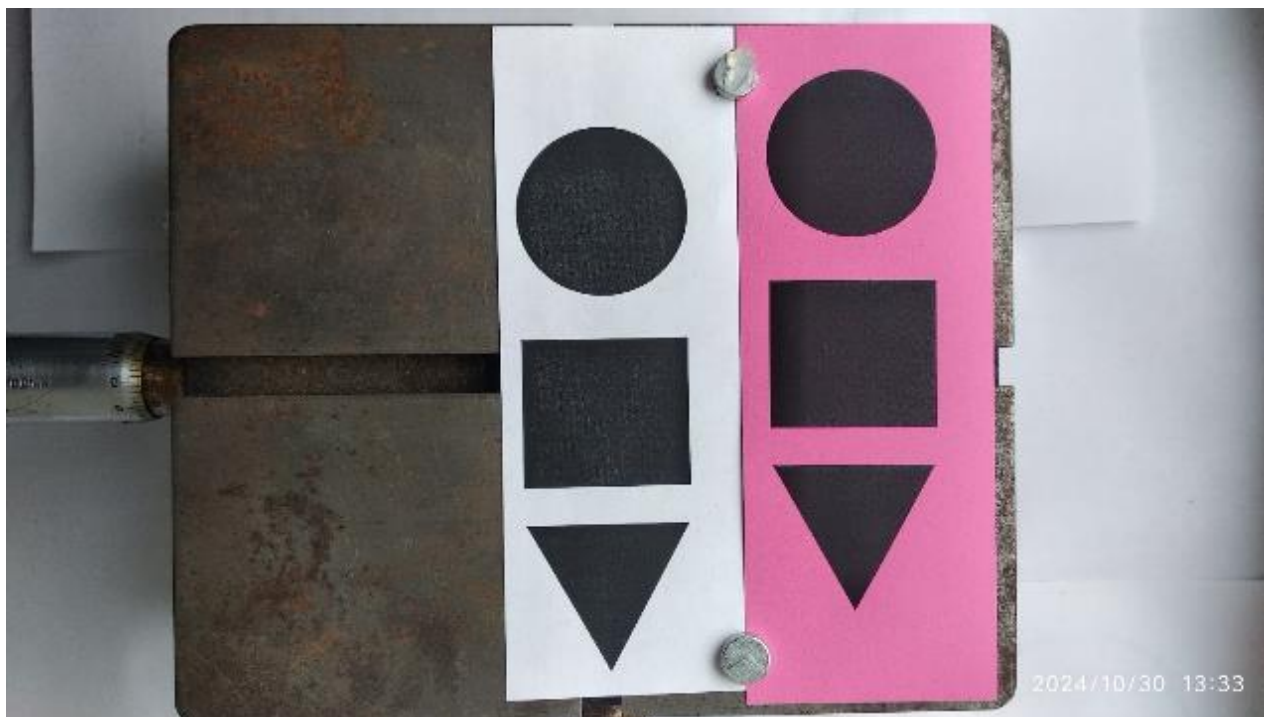


Рисунок 4.1 - Приклад фото для проведення тестів

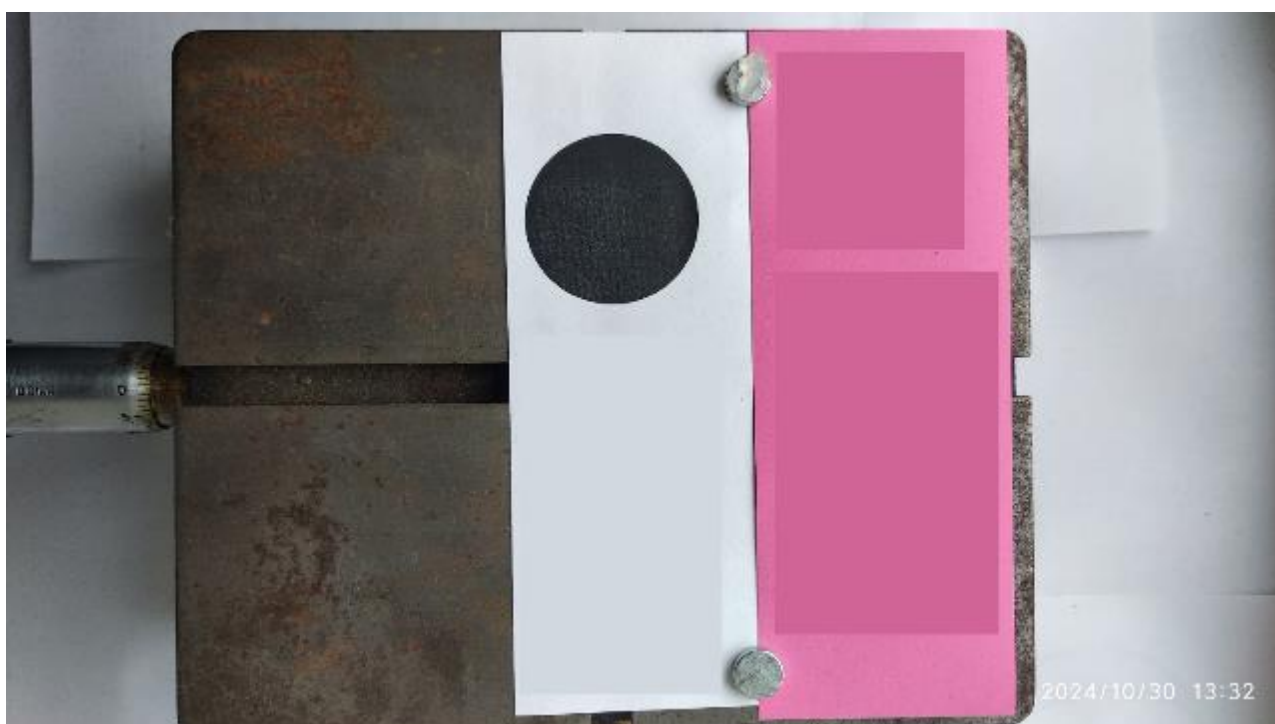


Рисунок 4.2 - Приклад ретушування зайвих геометричних фігур окрім кола на високо-контрастному фоні

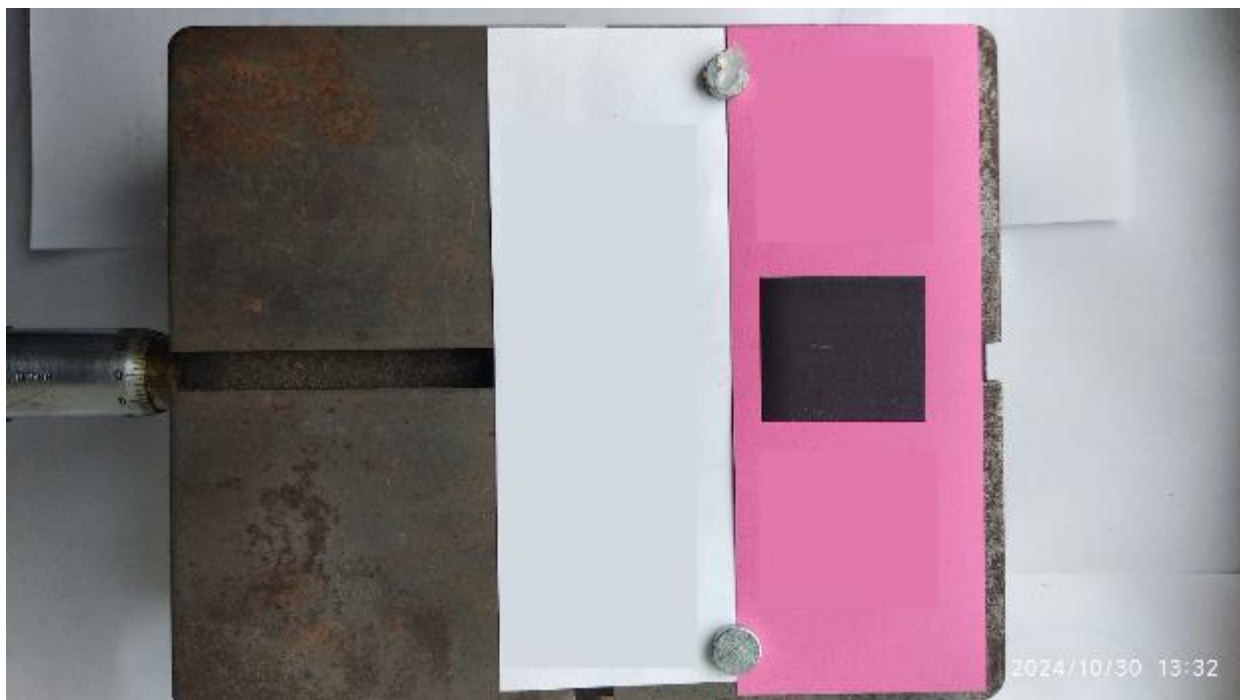


Рисунок 4.3 - Приклад ретушування зайвих геометричних фігур окрім квадрата на низько-контрастному фоні

4.2 Порівняння розробленого методу з іншими методами вимірювання

Було проведено тестування методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації, власне, результати аналізу представлено в таблиці 4.1.

Було проведено тестування іншого методу вимірювання визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації, результати якого можна побачити у таблиці 4.2.

Після отриманих результатів було сформовано графіки по кожній геометричній фігурі та фактичним значенням. На кожному графіку можна побачити наскільки точно той чи іншим метод визначив зміщення.

На кожному графіку по осі Y розміщене фактичне зміщення, що завжди дорівнює 0.5 мм, а по осі X розміщені зміщення між порівнюваними зображеннями.

Наприклад 1-2 по осі X показує зміщення між 1 та 2 зображенням.

Графіки розміщені на рисунках 4.4-4.14.

Таблиця 4.1 - Розрахункові зміщення маркерів отримано з використанням пропонованим методу

Новий метод	Коло на ВК	Квадрат на ВК	Трикутник на ВК	Коло на НК	Квадрат на НК	Трикутник на НК	Фактичне зміщення
1-2	0,50	0,52	0,49	0,51	0,50	0,49	0,50
2-3	0,47	0,49	0,46	0,52	0,47	0,51	0,50
3-4	0,42	0,48	0,51	0,52	0,42	0,48	0,50
4-5	0,48	0,46	0,52	0,51	0,48	0,49	0,50
5-6	0,56	0,49	0,48	0,49	0,56	0,51	0,50
6-7	0,51	0,54	0,45	0,48	0,52	0,48	0,50
7-8	0,57	0,53	0,48	0,5	0,51	0,46	0,50
8-9	0,59	0,49	0,47	0,5	0,54	0,55	0,50
9-10	0,51	0,47	0,51	0,46	0,48	0,52	0,50
10-11	0,42	0,45	0,46	0,52	0,46	0,46	0,50
11-12	0,47	0,51	0,52	0,51	0,50	0,49	0,50
12-13	0,46	0,52	0,54	0,49	0,53	0,48	0,50
13-14	0,47	0,49	0,52	0,50	0,48	0,50	0,50
14-15	0,50	0,51	0,47	0,51	0,50	0,47	0,50
15-16	0,43	0,53	0,48	0,47	0,49	0,49	0,50
16-17	0,46	0,48	0,50	0,45	0,51	0,49	0,50
17-18	0,55	0,51	0,52	0,48	0,52	0,51	0,50
18-19	0,54	0,48	0,51	0,52	0,51	0,50	0,50
Математичне сподівання	0,495	0,497	0,494	0,497	0,499	0,493	0,500
Дисперсія	0,0026	0,0006	0,0007	0,0005	0,0010	0,0005	-
Середнє квадратичне відхилення	0,103	0,051	0,053	0,043	0,064	0,045	-

Коефіцієнт варіації	0,103	0,051	0,053	0,043	0,064	0,045	-
Ступінь розпізнання	100%	100%	100%	100%	100%	100%	100%

Таблиця 4.2 - Розрахункові зміщення маркерів з використання методу на основі фільтрів

Фільтраційни й метод	Коло на ВК	Квадра т на ВК	Трикутні к на ВК	Коло на НК	Квадра т на НК	Трикутні к на НК	Фактичн е зміщенн я
1-2	0,25	0,75	-	1,5	0,50	0,60	0,50
2-3	-	0,20	1,80	0,65	-	-	0,50
3-4	0,15	-	0,35	0,80	1,20	0,70	0,50
4-5	0,30	0,85	0,25	0,25	0,90	1,75	0,50
5-6	-	1,50	-	0,75	-	0,60	0,50
6-7	0,10	-	-	1,80	0,20	-	0,50
7-8	0,20	0,70	0,90	0,60	1,00	-	0,50
8-9	-	0,15	-	-	-	1,50	0,50
9-10	0,30	1,80	0,20	-	0,65	-	0,50
10-11	1,50	-	-	0,20	-	0,60	0,50
11-12	-	-	0,25	1,20	0,35	-	0,50
12-13	-	0,60	-	0,75	-	0,30	0,50
13-14	0,20	-	-	-	-	1,80	0,50
14-15	-	-	0,50	1,50	-	0,75	0,50
15-16	0,35	-	-	-	0,80	-	0,50
16-17	-	0,25	-	-	1,00	0,60	0,50
17-18	0,40	-	-	0,9	-	-	0,50
18-19	0,10	-	0,75	-	-	1,50	0,50

Математичне сподівання	0,214	0,378	0,278	0,606	0,367	0,594	0,500
Дисперсія	0,1220	0,3048	0,2212	0,3553	0,1944	0,4132	-
Середнє квадратичне відхилення	1,633	1,461	1,693	0,984	1,203	1,081	-
Коефіцієнт варіації	1,633	1,461	1,693	0,984	1,203	1,081	-
Ступінь розпізнання	61%	50%	44%	66%	50%	61%	100%

Було розраховано математичне сподівання, дисперсію, середнє квадратичне відхилення та коефіцієнт варіації для нового методу та для інших методів, відповідні графіці продемонстровані на рисунках 4.15-4.22.

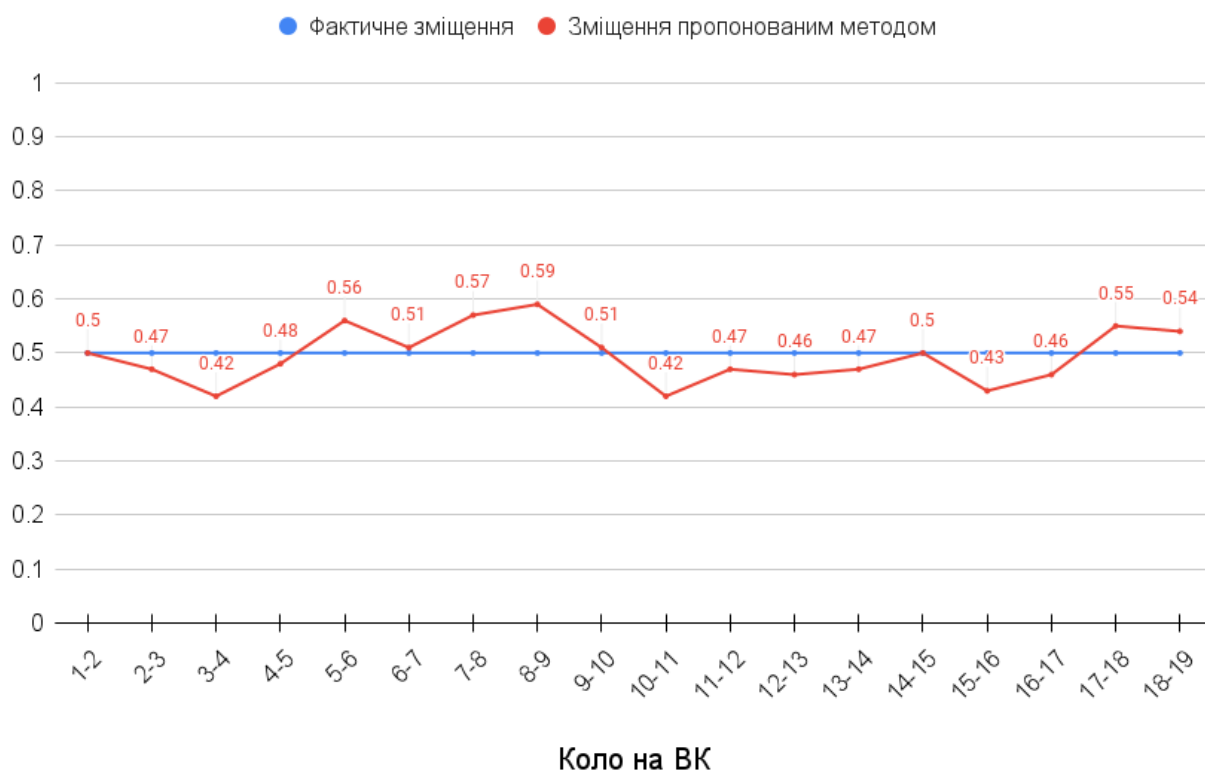


Рисунок 4.4 - Коло на високому контрасті у пропонованому методі



Рисунок 4.5 - Квадрат на високому контрасті у запропонованому методі



Рисунок 4.6 - Трикутник на високому контрасті у запропонованому методі

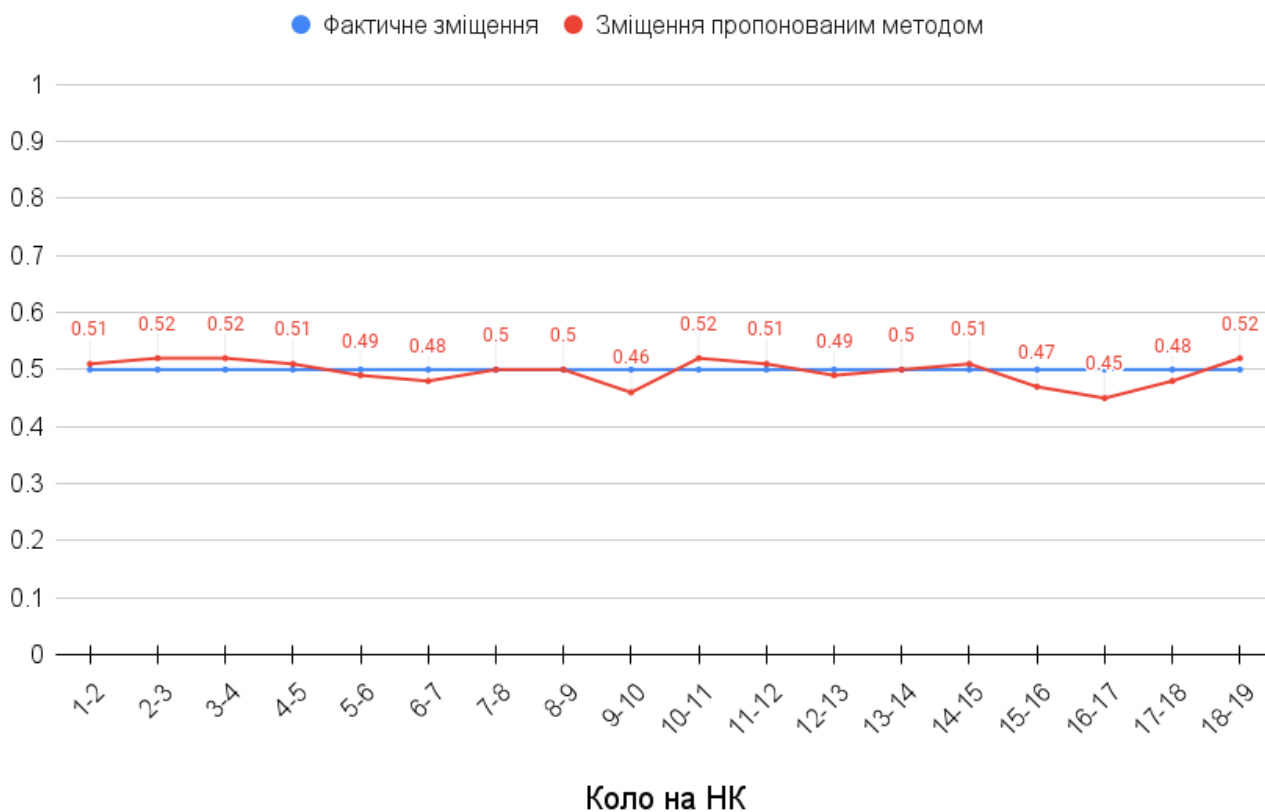


Рисунок 4.7 - Коло на низькому контрасті у пропонуваному методі



Рисунок 4.8 - Квадрат на низькому контрасті у пропонуваному методі

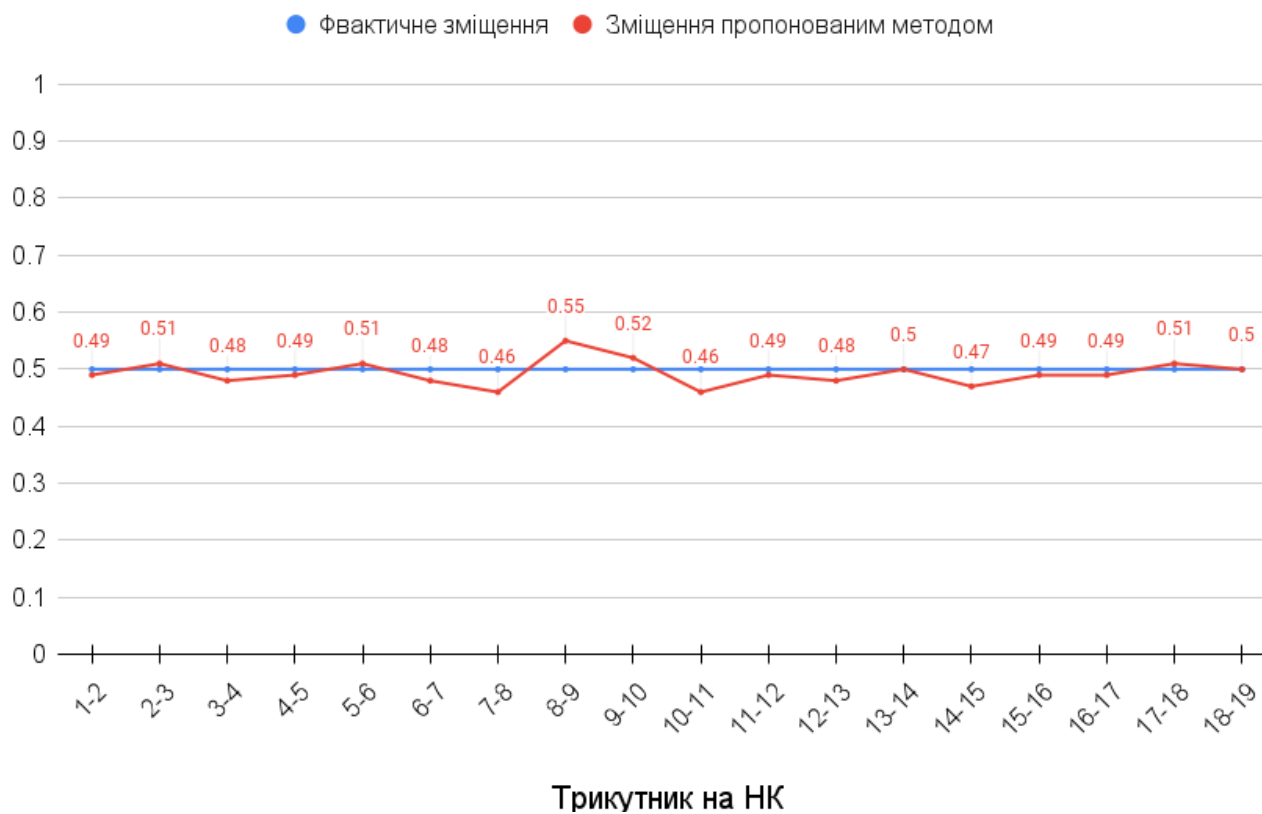


Рисунок 4.9 - Трикутник на низькому контрасті у запропонованому методі

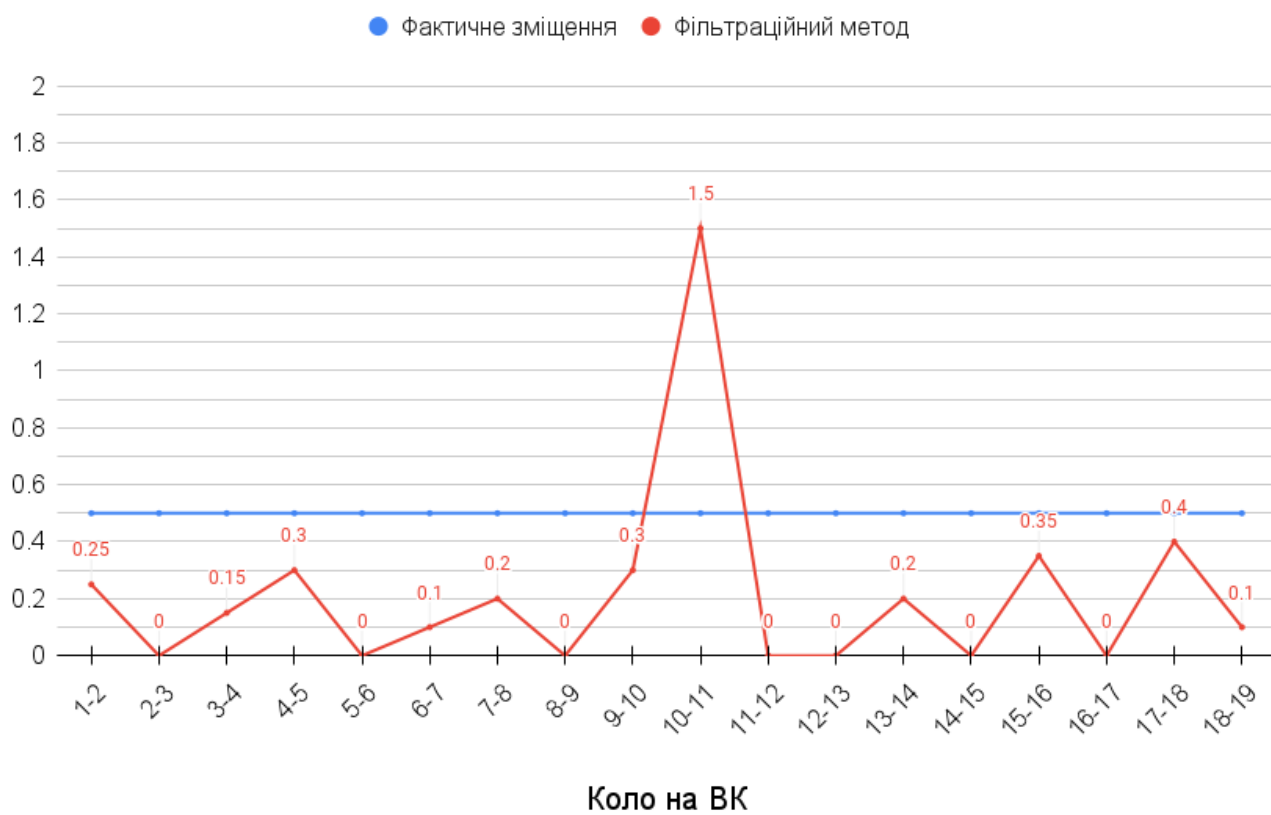


Рисунок 4.10 - Коло на високому контрасті у фільтраційному методі

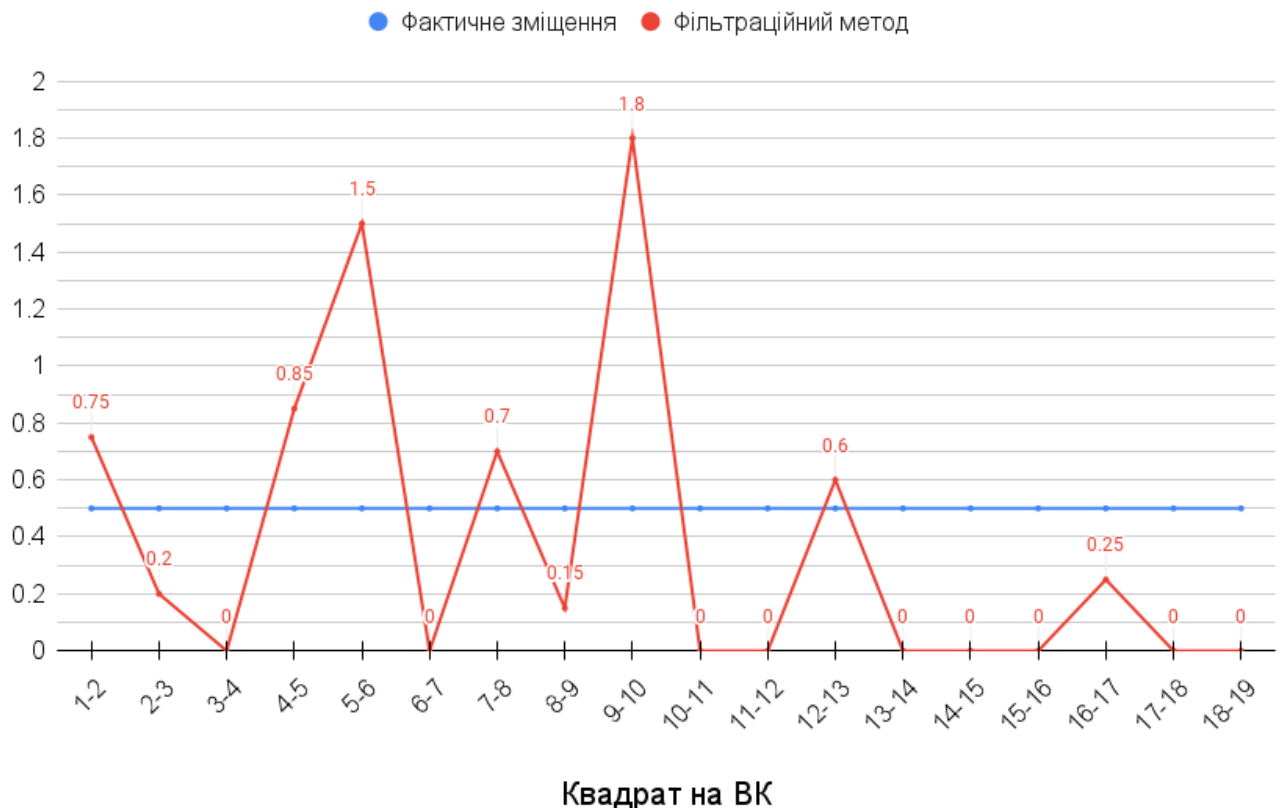


Рисунок 4.11 - Квадрат на високому контрасті у фільтраційному методі

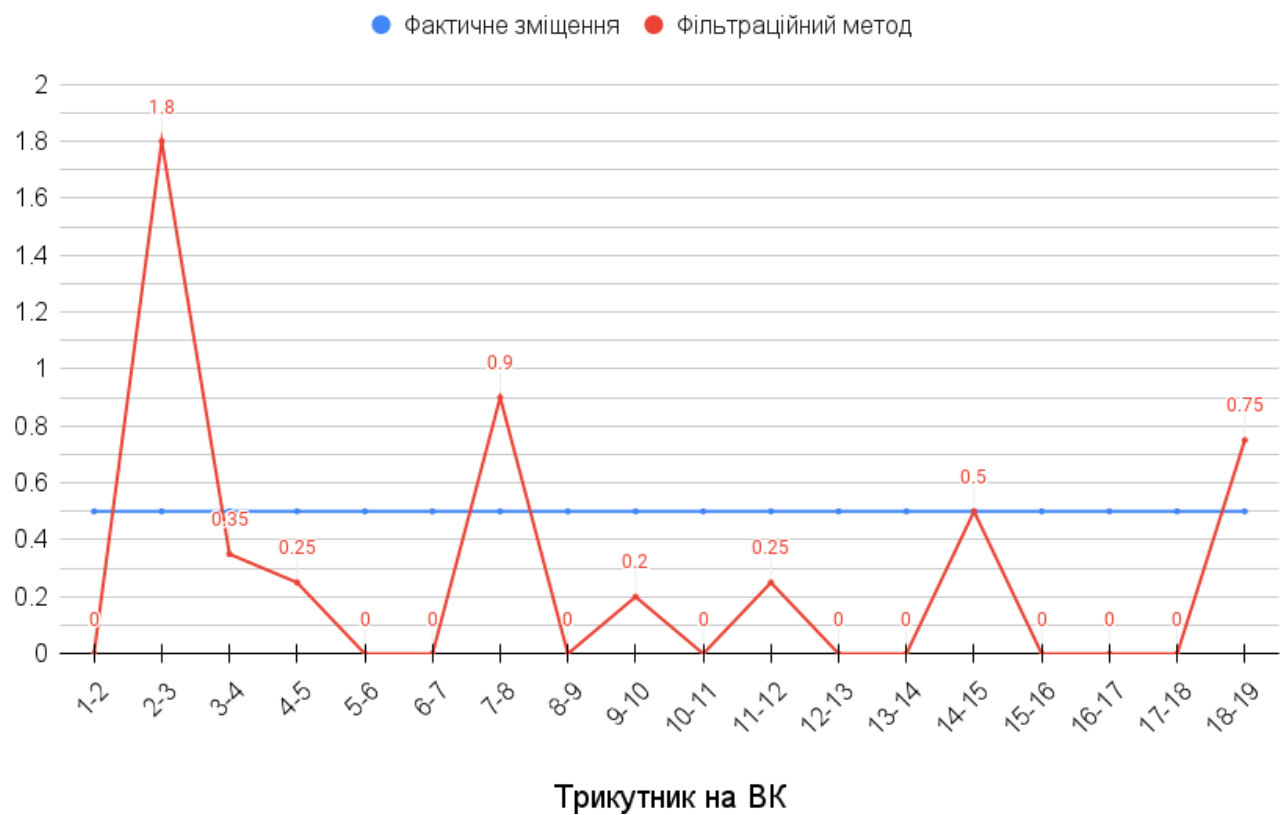


Рисунок 4.12 - Трикутник на високому контрасті у фільтраційному методі

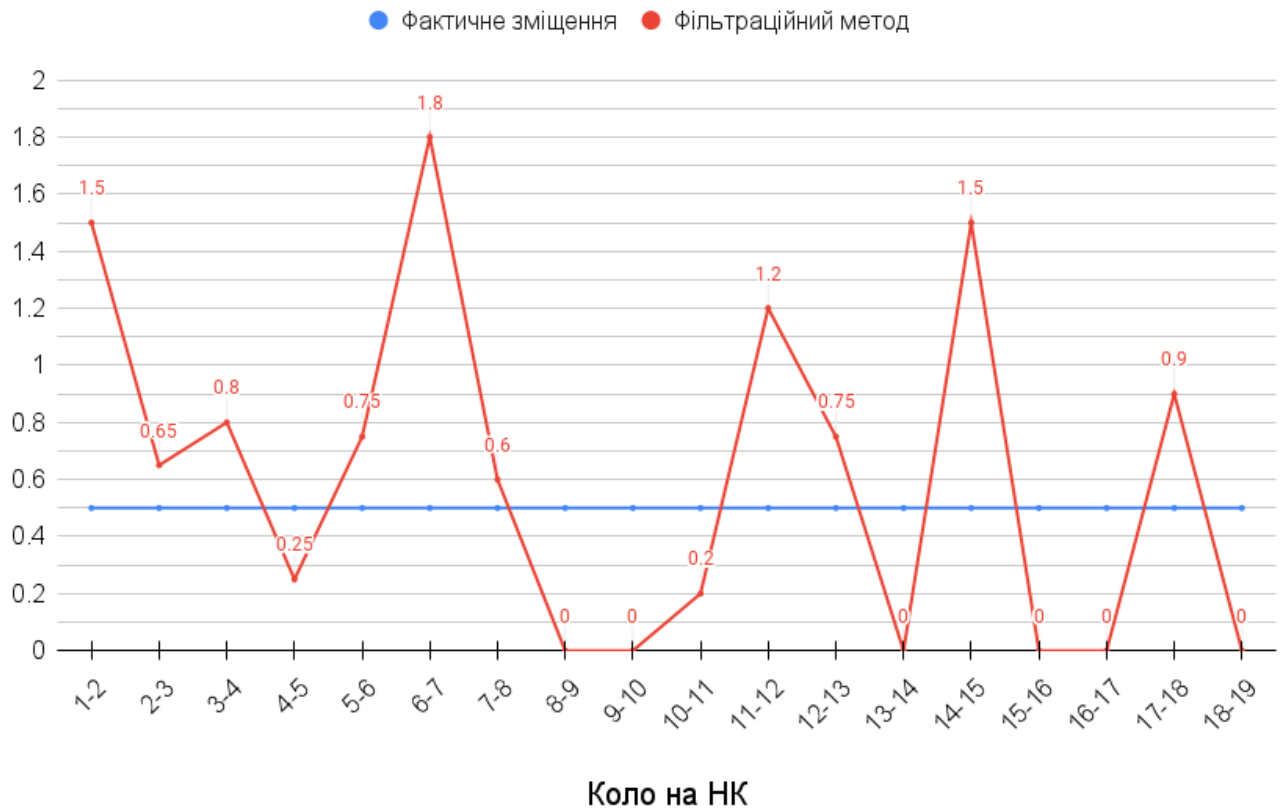


Рисунок 4.13 - Коло на низькому контрасті у фільтраційному методі

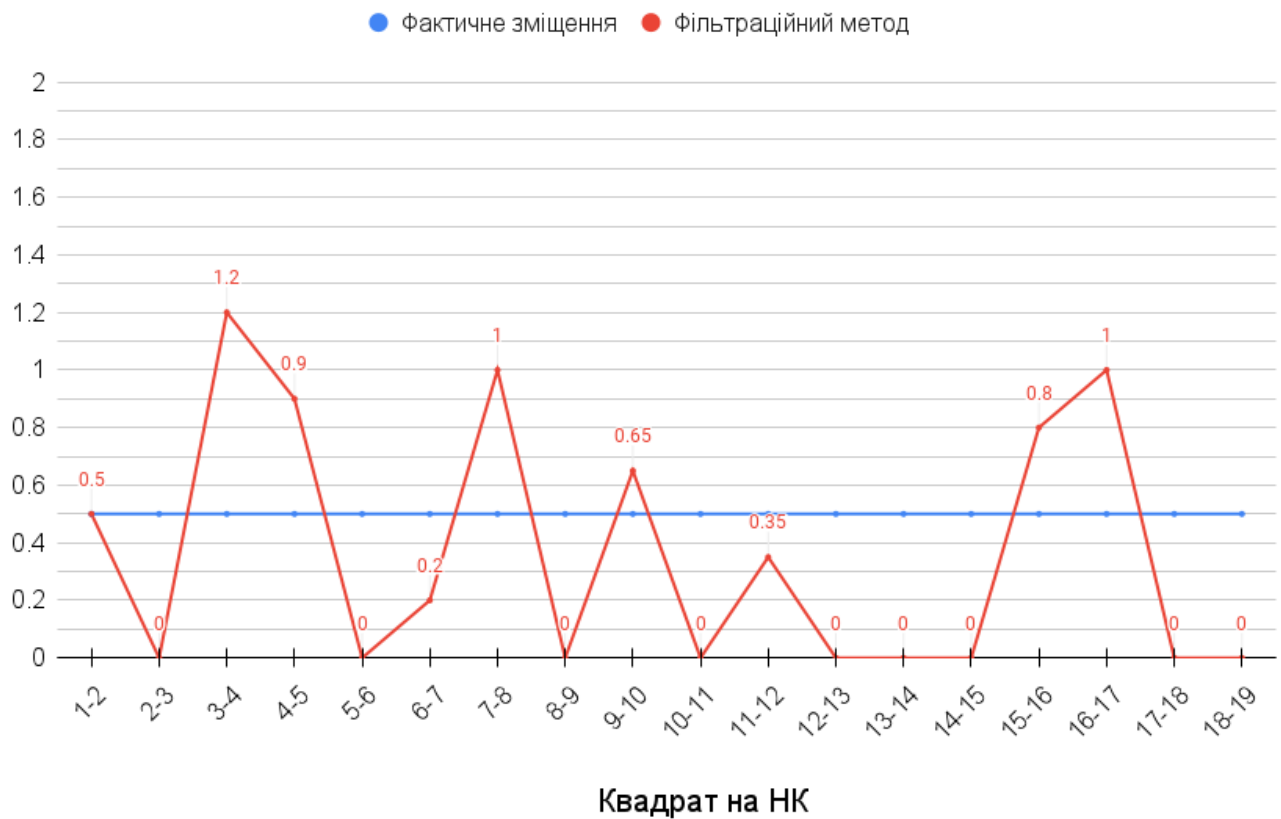


Рисунок 4.13 - Квадрат на низькому контрасті у фільтраційному методі

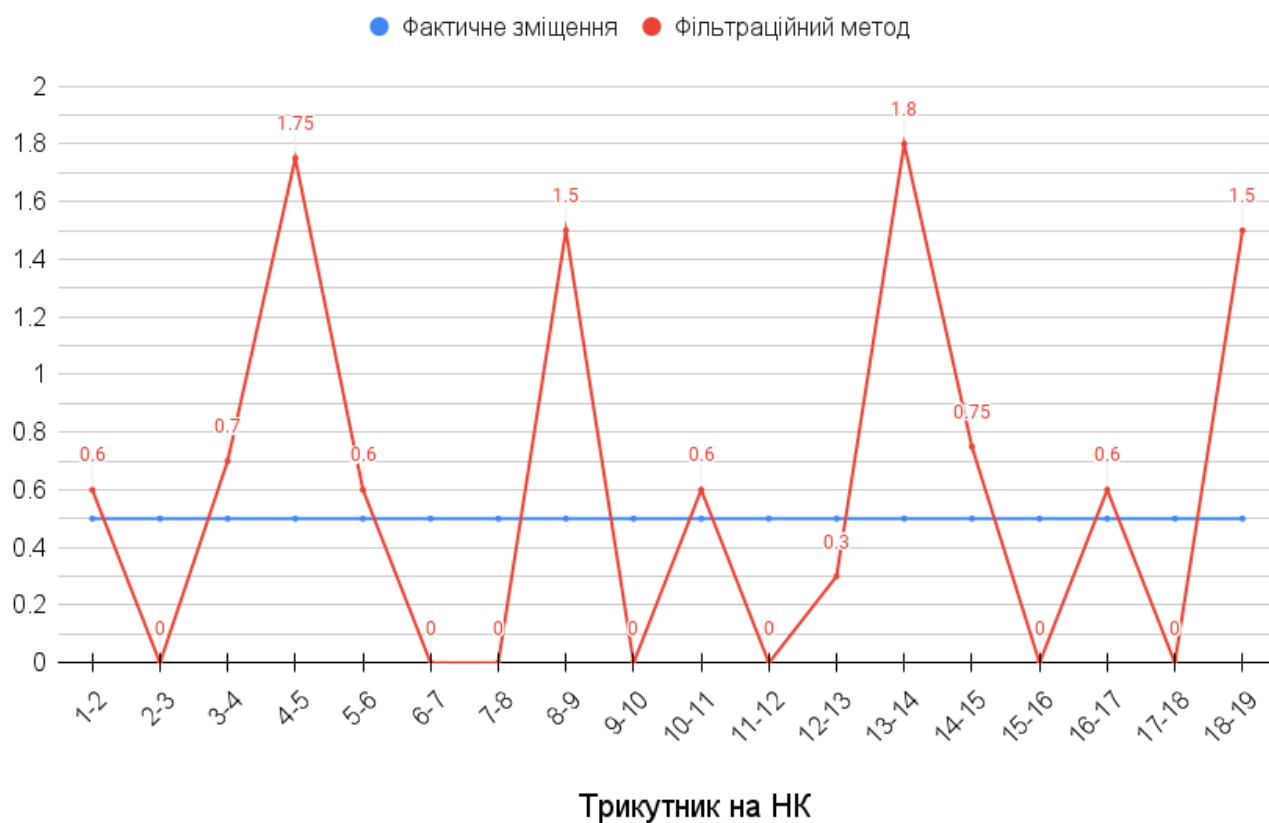


Рисунок 4.14 - Трикутник на низькому контрасті у фільтраційному методі



Рисунок 4.15 - Математичне сподівання у пропонованому методі



Рисунок 4.16 - Дисперсія у пропонованому методі



Рисунок 4.17 - Середнє квадратичне відхилення у пропонованому методі



Рисунок 4.18 - Коефіцієнт варіації у пропонованому методі

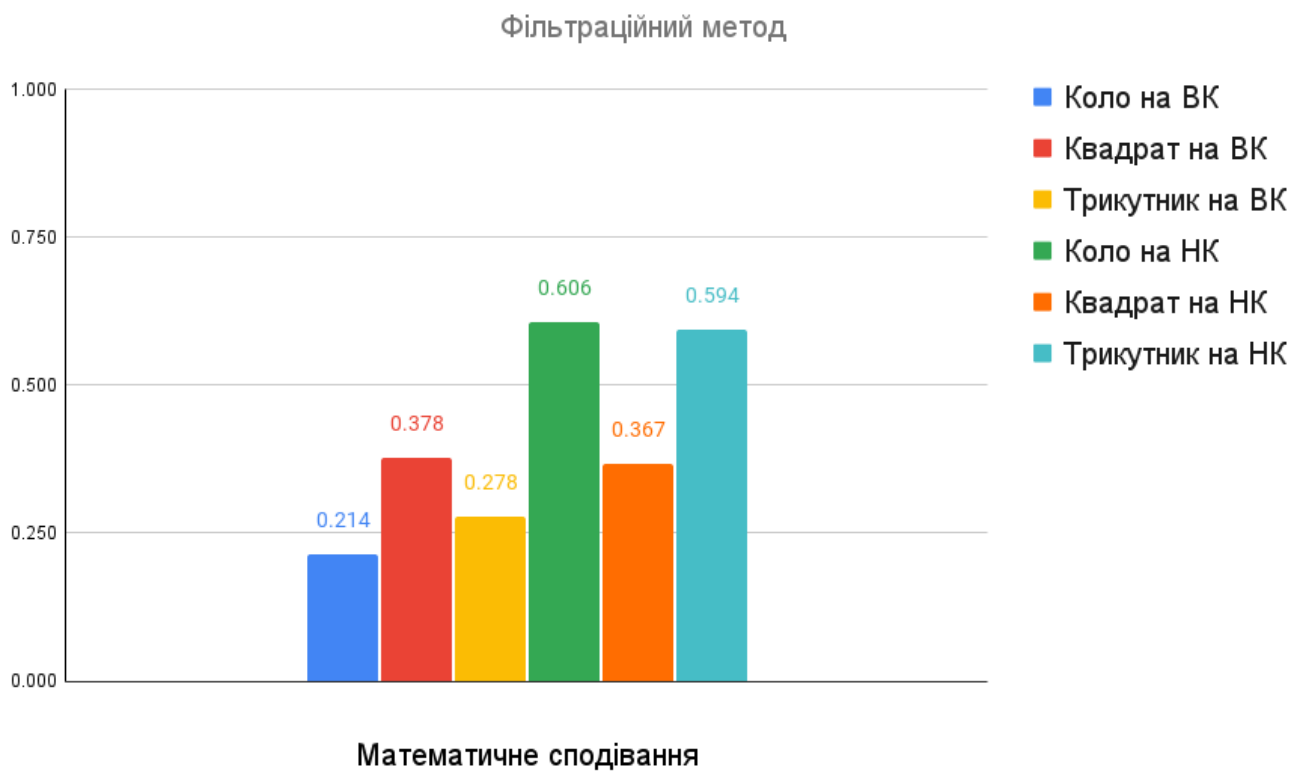


Рисунок 4.19 - Математичне сподівання у фільтраційному методі

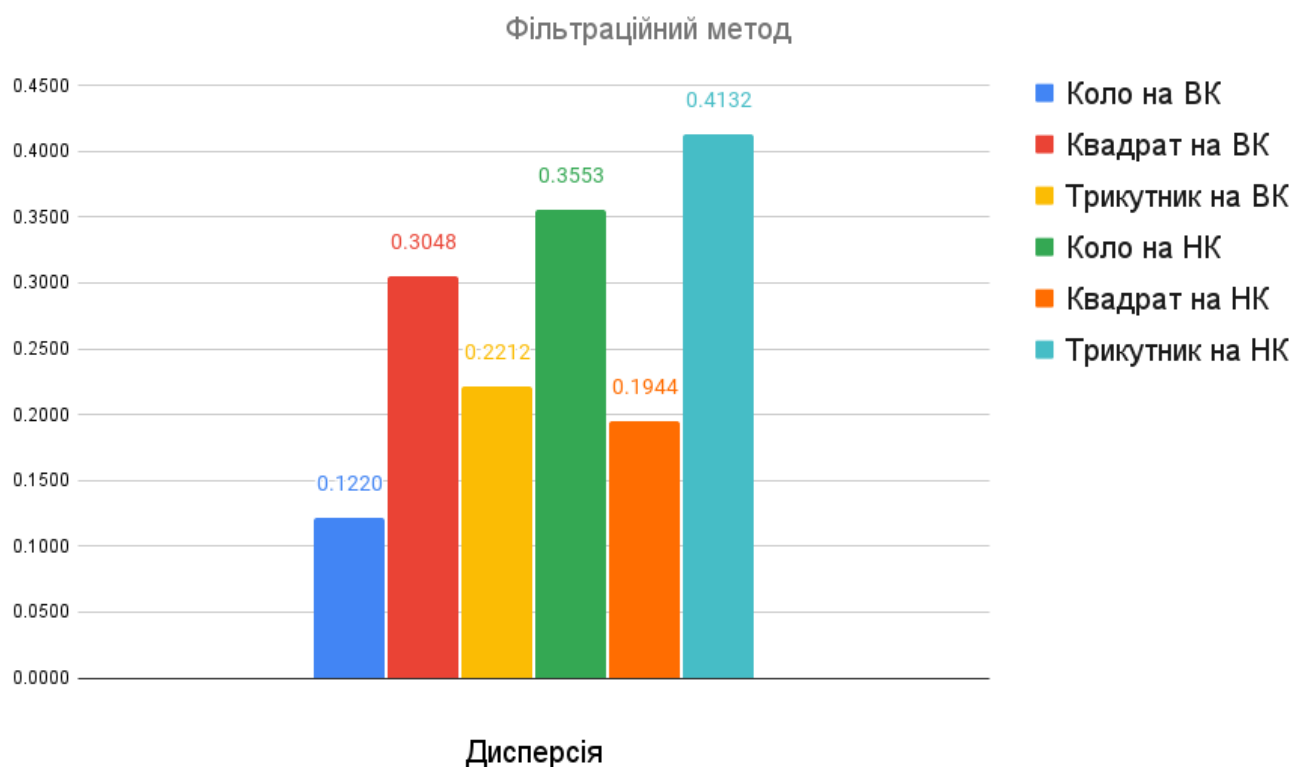


Рисунок 4.20 - Дисперсія у фільтраційному методі



Рисунок 4.21 - Середнє квадратичне відхилення у фільтраційному методі

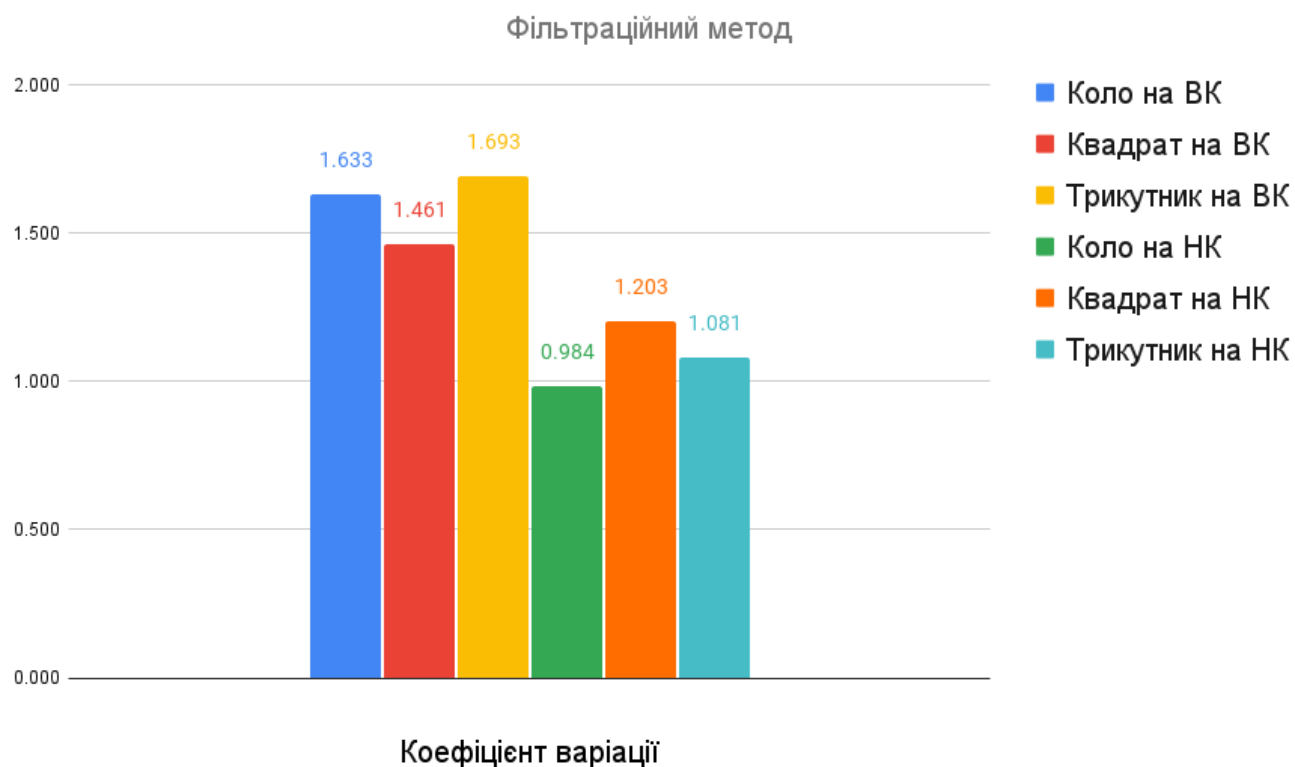


Рисунок 4.22 - Коефіцієнт варіації у фільтраційному методі

Після виконання усіх тестувань, було проведено порівняння між запропонованим методом та фільтраційним методом, результати порівняння наведено в таблиці 4.3.

У таблиці 4.3, у чисельника значення обчислена за запропонованим методом, а в знаменник значення обчислені за фільтраційним методом.

Таблиця 4.3 - Порівняння математичних ознак запропонованого методу з фільтраційним методом

	Математичне сподівання	Дисперсія	Середнє квадратичне відхилення	Коефіцієнт варіації	Ступінь розпізнання
Коло на ВК	0.495	0.0026	0.051	0.103	100.00%
	0.214	0.1220	0.349	1.633	61.11%
Квадрат на ВК	0.497	0.0006	0.025	0.051	100.00%
	0.378	0.3048	0.552	1.461	50.00%

Трикутник на ВК	0.494	0.0007	0.026	0.053	100.00%
	0.278	0.2212	0.470	1.693	44.44%
Коло на НК	0.497	0.0005	0.021	0.043	100.00%
	0.606	0.3553	0.026	0.053	100.00%
Квадрат на НК	0.499	0.0010	0.032	0.064	100.00%
	0.367	0.1944	0.441	1.203	50.00%
Трикутник на НК	0.493	0.0005	0.022	0.045	100.00%
	0.6	0.4132	0.643	1.081	61.11%

Відповідно до Таблиці 4.3 реалізовано графіки на рисунках 4.23-4.27 на яких візуально порівняно математичне сподівання, дисперсію, середнє квадратичне відхилення, коефіцієнт варіації та ступінь розпізнання.

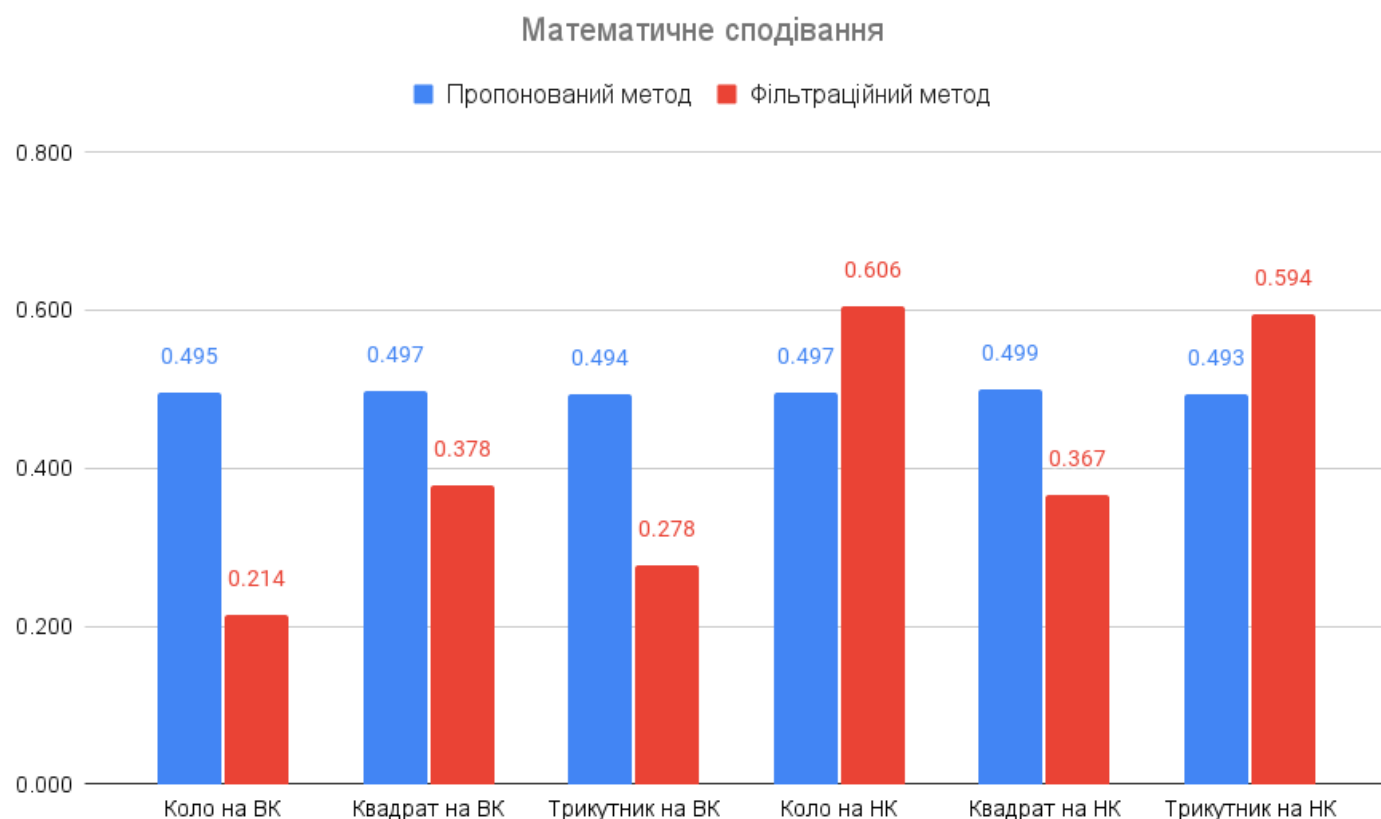


Рисунок 4.23 - Порівняння математичного сподівання запропонованого методу з фільтраційним методом

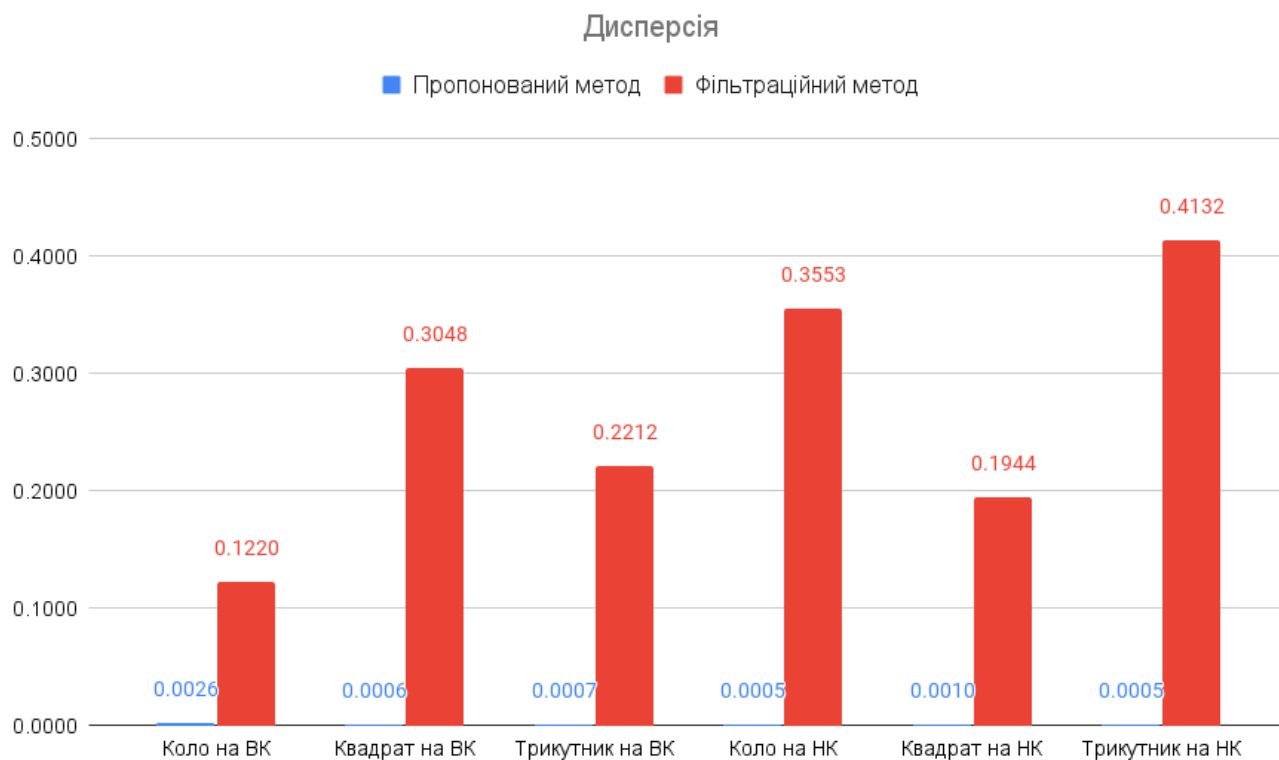


Рисунок 4.24 - Порівняння дисперсії запропонованого методу з фільтраційним методом



Рисунок 4.25 - Порівняння середнього квадратичного відхилення запропонованого методу з фільтраційним методом

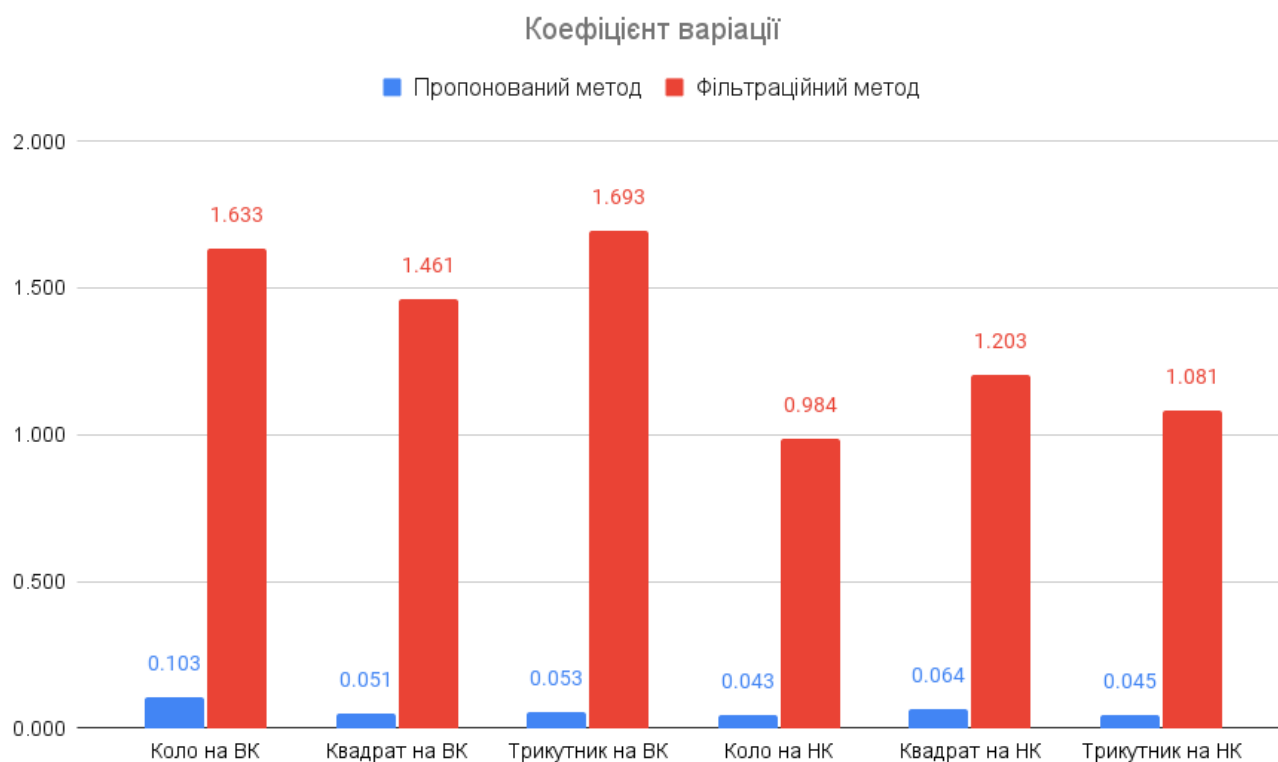


Рисунок 4.26 - Порівняння коефіцієнту варіації запропонованого методу з фільтраційним методом

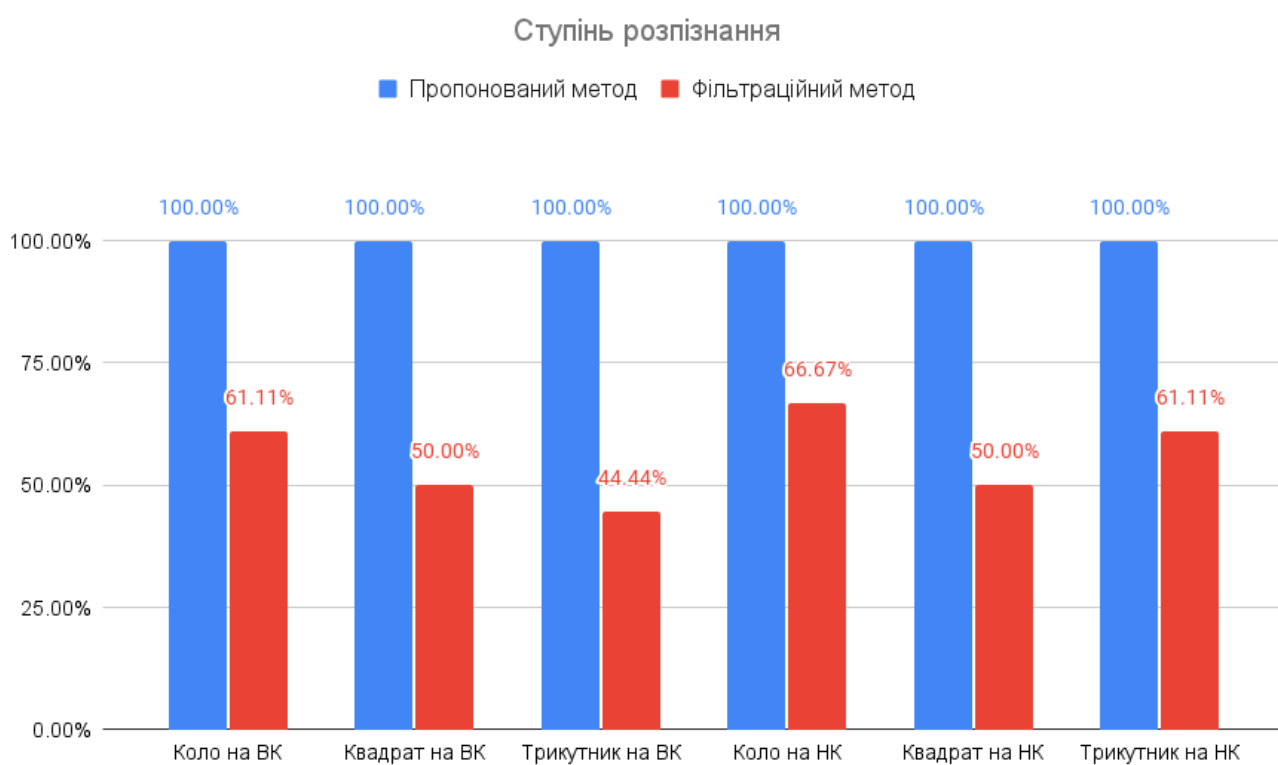


Рисунок 4.27 - Порівняння ступеня розпізнавання запропонованого методу з фільтраційним методом

Висновки до розділу 4

За результатами проведеного дослідження підтверджено суттєві переваги розробленого методу у порівнянні з традиційним підходом, які ґрунтуються на застосуванні графічних фільтрованим.

Результати запропонованого методу, його математичне сподівання має мінімальне значення 0,493 мм на трикутнику високого контрасту та максимальне значення на квадраті низького контрасту 0,499 мм при фактичному значенні 0,5 мм. Мінімальні значення дисперсії спостерігається у квадрата на високому контрасті 0,0006 мм², а максимальне значення є у кола на високому контрасті 0,0026 мм². Середнє квадратичне відхилення знаходиться в діапазоні від 0,021 мм до 0,053 мм, з мінімальним значенням для кола на низькому контрасті 0,021 мм та максимальним значенням для кола на високому контрасті з значенням 0,053 мм. Коефіцієнт варіації має мінімальне значення для кола на низькому контрасті 0,043, а максимальне значення для кола на високому контрасті 0,103. Метод має ступеня розпізнання 100% для всіх об'єктів і контрастних умов.

При визначені величини зміщення без використання методу кластеризації математичне сподівання становило для кола на високому контрасті 0,214 мм (мінімальне значення) та у кола на низькому контрасті 0,606 мм (максимальне значення) при фактичному значенні 0,5 мм. Максимальне значення дисперсії має трикутник на низькому контрасті 0,4132 мм², а мінімальне значення має коло на високому контрасті 0,1220 мм². Середнє квадратичне відхилення досягає максимального значення 1,693 для трикутника на високому контрасті і мінімального значення 0,349 для кола на високому контрасті. Коефіцієнт варіації коливається від 0,984 для кола на низькому контрасті до 1,693 для трикутника на високому контрасті. Мінімальне значення ступеня розпізнання є трикутник на високому контрасті 44,44% та максимальне значення 66,67% для кола на низькому контрасті.

Таким чином, відхилення середнього значення величини зміщення отримане під час експериментальної апробації методу від фактичного не перевищувало ~2%, в той час як для порівнюваного методу це ж відхилення не спостерігалось меншим за

20%. Значення середньоквадратичного відхилення для запропонованого методу мають значення одного порядку із величиною точності фізичного приладу, що використовувався під час практичного тестування методу, в той час як за умови невикористання методів кластеризації при визначенні контурів маркеру мають місце суттєво гірші показники.

Загальні висновки

Під час виконання кваліфікаційної роботи магістра було запропоновано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням кластеризації.

Було розроблено метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням кластеризації. Підвищення точності визначення величини зміщення об'єкту на зображеннях досягається за рахунок використання методу кластеризації k-середніх при визначенні контурів маркеру, переміщення якого приймаються ідентичними переміщенням об'єкту спостереження.

Побудовано загальну структуру методу підвищення точності визначення зміщення об'єктів з адаптацією та імплементацією методу k-середніх з покроковою його деталізацією.

Сегментація зображень виконувалася з використанням методу k-середніх. Автоматизовано сегментацію зображень у поєднанні з фільтрацією шумів, підвищенням контрастності та нормалізацією.

Було реалізовано поставлені задачі:

- проведено аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектовано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначено набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконано програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначено точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.

Виконано програмну реалізацію методу підвищення точності визначення зміщення об'єктів за допомогою кластеризації.

Програмна реалізація виконана на основі архітектури системи з 8 модулів, серед яких завантаження зображення, попередня обробка, кластеризація, виявлення об'єктів, аналіз зміщення, візуалізація, управління та логування на мові програмування Python та за допомогою OpenCV, scikit-learn, Tkinter, PIL, skimage.

Виконано експериментальне тестування запропонованого методу на серії зображень для трьох форм маркерів чорного кольору, які штучним чином наносилися на висококонтрастний, що маркера, (білий) та низькоконтрастний (рожевий) фон поверхні.

Експериментальна перевірка довела підвищення точності вимірювання переміщення за аналізом зображень з використанням методів кластеризації. Так відхилення середнього значення величини зміщення отримане під час експериментальної апробації методу від фактичного не перевищувало ~2%, в той час як для порівнюваного методу це ж відхилення не спостерігалось меншим за 20%. Значення середньоквадратичного відхилення для запропонованого методу мають значення одного порядку із величиною точності фізичного приладу, що використовувався під час практичного тестування методу, в той час як за умови невикористання методів кластеризації при визначенні контуру маркера мають місце суттєво гірші показники.

Перелік посилань

1. Singh A. What is SIFT(Scale Invariant Feature Transform) Algorithm?. analyticsvidhya. URL: [https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=The%20SIFT%20\(Scale-Invariant%20Feature,%,%20rotation,%20and%20affine%20transformations](https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=The%20SIFT%20(Scale-Invariant%20Feature,%,%20rotation,%20and%20affine%20transformations) (date of access: 11.11.2024).
2. Deep. Introduction to SURF (Speeded-Up Robust Features). medium. URL: <https://medium.com/@deepanshut041/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e> (date of access: 11.11.2024).
3. Awaits. [CV] 2. Gaussian and Median Filter, Separable 2D filter. Medium. URL: <https://medium.com/jun94-devpblog/cv-2-gaussian-and-median-filter-separable-2d-filter-2d11ee022c66> (date of access: 11.11.2024).
4. Sahir S. Canny edge detection step by step in python – computer vision. Medium. URL: <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123?gi=f0e0be9dcc47> (date of access: 11.11.2024).
5. Malche T. Edge Detection in Image Processing: An Introduction. Roboflow Blog. URL: <https://blog.roboflow.com/edge-detection/#:~:text=Sobel%20edge%20detection%20is%20a,each%20pixel%20in%20the%20image> (date of access: 11.11.2024).
6. Algorithm Selection for Edge Detection in Satellite Images by Neutrosophic WASPAS Method. MDPI. URL: <https://www.mdpi.com/2071-1050/12/2/548> (date of access: 11.11.2024).
7. Chhangani H. Pattern Matching Algorithm. Medium. URL: <https://medium.com/@harshitachhangani0503/pattern-matching-algorithm-4ca950792c95> (date of access: 11.11.2024).
8. Correlation Coefficient. URL: https://www.jmp.com/en_ca/statistics-knowledge-portal/what-is-correlation/correlation-coefficient.html#:~:text=The%20correlation%20coefficient%20is%20the,r%20in%20a%20correlation%20report (date of access: 25.11.2024).

9. Keita Z. An Introduction to Convolutional Neural Networks (CNNs). datacamp. URL: <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns> (date of access: 11.11.2024).
10. Boesch G. Very Deep Convolutional Networks (VGG) Essential Guide - viso.ai. viso.ai. URL: <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/> (date of access: 11.11.2024).
11. Azeem I. Understanding ResNet Architecture: A Deep Dive into Residual Neural Network. Medium. URL: <https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9> (date of access: 11.11.2024).
12. Aranda Diaz A. Agile Inception. Answering the questions What, Why and How?-EN. Netmind. URL: <https://netmind.net/en/agile-inception-answering-the-questions-what-why-and-how-en/> (date of access: 11.11.2024).
13. Kundu R. YOLO Algorithm for Object Detection Explained [+Examples]. V7 | AI Document Processing & Data Labelling. URL: <https://www.v7labs.com/blog/yolo-object-detection> (date of access: 11.11.2024).
14. Pujara A. Object segmentation with Mask R-CNN. Medium. URL: <https://abhijeetpujara.medium.com/object-segmentation-with-mask-r-cnn-6e9c06cef81c> (date of access: 11.11.2024).
15. Thakur V. K. Faster R-CNN : Object Detection. Medium. URL: <https://medium.com/thedeephub/faster-r-cnn-object-detection-5dfe77104e31> (date of access: 11.11.2024).
16. Support Vector Machine (SVM) Algorithm - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/support-vector-machine-algorithm/> (date of access: 11.11.2024)
17. T. Holloway S. modelObj: A Model Object Framework for Regression Analysis. The Comprehensive R Archive Network. URL: <https://cran.r-project.org/web/packages/modelObj/vignettes/modelObj.pdf> (date of access: 11.11.2024).

18. Donges N. Random Forest: A Complete Guide for Machine Learning | Built In. Built In. URL: <https://builtin.com/data-science/random-forest-algorithm> (date of access: 11.11.2024).
19. CR A. Exploring Clustering Algorithms: Explanation and Use Cases. neptune.ai. URL: <https://neptune.ai/blog/clustering-algorithms> (date of access: 11.11.2024)
20. Sachinoni. The Art and Science of K-means Clustering: A Practical Guide. Medium. URL: <https://medium.com/@sachinoni600517/the-art-and-science-of-k-means-clustering-a-practical-guide-e71b11638867> (date of access: 11.11.2024).
21. Sachinoni. Clustering Like a Pro: A Beginner's Guide to DBSCAN. Medium. URL: <https://medium.com/@sachinoni600517/clustering-like-a-pro-a-beginners-guide-to-dbscan-6c8274c362c4> (date of access: 11.11.2024).
22. Noble J. What is Hierarchical Clustering? | IBM. IBM - United States. URL: <https://www.ibm.com/think/topics/hierarchical-clustering#:~:text=Hierarchical%20clustering%20is%20an%20unsupervised,patterns%20and%20connections%20in%20datasets> (date of access: 11.11.2024).
23. Jain R. Implementing Spectral Clustering from Scratch: A Step-by-Step Guide. Medium. URL: <https://rahuljain788.medium.com/implementing-spectral-clustering-from-scratch-a-step-by-step-guide-9643e4836a76> (date of access: 11.11.2024).
24. Dhumne S. Mean-Shift Clustering: A Powerful Technique for Data Analysis with Python. Medium. URL: <https://medium.com/@shruti.dhumne/mean-shift-clustering-a-powerful-technique-for-data-analysis-with-python-f0c26bfb808a> (date of access: 11.11.2024).
25. Sahoo S. An Analysis of Normalized Cuts and Image Segmentation. Medium. URL: <https://medium.com/analytics-vidhya/an-analysis-of-normalized-cuts-and-image-segmentation-1acb05d1a2e9> (date of access: 11.11.2024).
26. Huang C. Polarization measurement accuracy analysis and improvement methods for the directional polarimetric camera. optica. URL: <https://opg.optica.org/oe/fulltext.cfm?uri=oe-28-26-38638&id=444366> (date of access: 11.11.2024).

27. eelectronics. The Evolution of Sensor Technology: From Analog to Digital. Medium. URL: <https://eelectronics.medium.com/the-evolution-of-sensor-technology-from-analog-to-digital-3857b6f37629> (date of access: 11.11.2024).
28. Kotsonis L. What Is Sensor Calibration And Why It Matters. Bota Systems. URL: <https://www.botasys.com/post/sensor-calibration> (date of access: 11.11.2024).
29. Morgan T. Implementing a Kalman Filter for Better Noise Filtering. Losant Enterprise IoT Platform. URL: <https://www.losant.com/blog/implementing-a-kalman-filter-for-better-noise-filtering> (date of access: 11.11.2024).
30. Khadir M. T. A multiple clustering combination approach based on iterative voting process. sciencedirect. URL: <https://www.sciencedirect.com/science/article/pii/S131915781930597X> (date of access: 11.11.2024).
31. Sharma N. K-Means Clustering Explained. neptune.ai. URL: <https://neptune.ai/blog/k-means-clustering> (date of access: 11.11.2024).
32. Sarker I. H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. researchgate. URL: https://www.researchgate.net/publication/353986944_Deep_Learning_A_Comprehensive_Overview_on_Techniques_Taxonomy_Applications_and_Research_Directions (date of access: 11.11.2024).
33. Contributors to Wikimedia projects. Python (programming language) - Wikipedia. Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (date of access: 11.11.2024).
34. OpenCV Tutorial in Python - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/opencv-python-tutorial/> (date of access: 11.11.2024).
35. Gupta A. An Introduction to Scikit-Learn: Machine Learning in Python. Simplilearn.com. URL: <https://www.simplilearn.com/tutorials/python-tutorial/scikit-learn> (date of access: 11.11.2024).

36. Громко Г. Знайомимось з Tkinter. Medium. URL: <https://medium.com/@gr.gromko/знайомимось-з-tkinter-c89ee588234c> (дата звернення: 11.11.2024).

37. GeeksforGeeks. Python: Pillow (a fork of PIL) - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/python-pillow-a-fork-of-pil/> (date of access: 11.11.2024).

38. Singh A. 9 Powerful Tips and Tricks for Working with Image Data using skimage in Python. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2019/09/9-powerful-tricks-for-working-image-data-skimage-python/> (date of access: 11.11.2024).

39. Convert BGR and RGB with Python, OpenCV (cvtColor) | note.nkmk.me. nkmk note. URL: <https://note.nkmk.me/en/python-opencv-bgr-rgb-cvtColor/> (date of access: 11.11.2024).

40. GeeksforGeeks. Create Local Binary Pattern of an image using OpenCV-Python - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/create-local-binary-pattern-of-an-image-using-opencv-python/> (date of access: 11.11.2024).

ДОДАТКИ

Додаток А

main.py

```

import cv2
import numpy as np
from tkinter import (
    Text, Tk, Label, Button, filedialog, messagebox,
    Scrollbar, LEFT, RIGHT, Y, Scale, HORIZONTAL, Frame, END, Entry, Checkbutton, IntVar
)
from tkinter.ttk import Progressbar
from PIL import Image, ImageTk
import threading
import queue
import os
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from scipy.optimize import linear_sum_assignment
from skimage.feature import local_binary_pattern

RESULTS_FOLDER = "results"

KNOWN_SHIFT_MM_DEFAULT = 0.5

def preprocess_image(image, resize_width=800, kernel_size=(5, 5)):
    height, width = image.shape[:2]
    if width > resize_width:
        scaling_factor = resize_width / float(width)
        image = cv2.resize(image, None, fx=scaling_factor, fy=scaling_factor,
interpolation=cv2.INTER_AREA)

    lab = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)
    blurred = cv2.GaussianBlur(lab, kernel_size, 0)
    lab[:, :, 0] = cv2.equalizeHist(blurred[:, :, 0])

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)

    return lab, image, thresh

def extract_texture_features_lbp(image_gray, radius=1, n_points=8):
    lbp = local_binary_pattern(image_gray, n_points, radius, method='uniform')
    (hist, _) = np.histogram(lbp.ravel(),
        bins=np.arange(0, n_points + 3),
        range=(0, n_points + 2))

    hist = hist.astype("float")
    hist /= (hist.sum() + 1e-7)

```

```

return hist

def optimal_k(data, max_k=10):
    scores = []
    for k in range(2, max_k + 1):
        kmeans = KMeans(n_clusters=k, random_state=42)
        labels = kmeans.fit_predict(data)
        if len(np.unique(labels)) < 2:
            scores.append(-1)
            continue
        score = silhouette_score(data, labels)
        scores.append(score)
    optimal_k_value = scores.index(max(scores)) + 2 if scores else 2
    print(f"Optimal k determined: {optimal_k_value}")
    return optimal_k_value

def kmeans_clustering(image, k=None):
    height, width = image.shape[:2]
    X, Y = np.meshgrid(np.arange(width), np.arange(height))
    X = X / width
    Y = Y / height
    Z = np.dstack((image, X, Y)).reshape((-1, 5))
    Z = np.float32(Z)

    if k is None:
        k = optimal_k(Z, max_k=10)

    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(Z)
    labels = kmeans.labels_
    centers = kmeans.cluster_centers_.astype("uint8")

    centers_colors = centers[:, :3]

    segmented_image = centers_colors[labels.flatten()].reshape((image.shape))

    print(f"Number of clusters: {k}")
    print(f"Cluster centers:\n{centers}")

    return segmented_image, labels.flatten(), centers_colors

def is_black(color_lab):
    L, a, b = color_lab
    L_threshold = 50
    A_threshold = 80
    B_threshold = 80

```

```

return (L < L_threshold) and (abs(a - 128) < A_threshold) and (abs(b - 128) < B_threshold)

def calculate_circularity(contour):
    perimeter = cv2.arcLength(contour, True)
    area = cv2.contourArea(contour)
    if perimeter == 0:
        return 0
    circularity = 4 * np.pi * (area / (perimeter * perimeter))
    return circularity

def calculate_vertices(contour):
    epsilon = 0.04 * cv2.arcLength(contour, True)
    approx = cv2.approxPolyDP(contour, epsilon, True)
    return len(approx)

def determine_shape(vertices, circularity, aspect_ratio=1.0):
    if 2 <= vertices <= 4:
        if vertices == 3:
            return 'Triangle'
        elif vertices == 4:
            if 0.8 <= aspect_ratio <= 1.2:
                return 'Square'
    if circularity > 0.7:
        return 'Circle'
    return None

def calculate_centroid(contour):
    M = cv2.moments(contour)
    if M['m00'] == 0:
        return (0, 0)
    cx = M['m10'] / M['m00']
    cy = M['m01'] / M['m00']
    return (cx, cy)

def extract_clusters(segmented_image, labels, centers, image_gray, min_cluster_size=1000,
                    aspect_ratio_threshold=1.3, circularity_threshold=0.6, kernel_size=(3, 3), iterations=1, pair_id=1):
    clusters = []
    height, width = segmented_image.shape[:2]
    black_clusters_count = 0
    for i, center in enumerate(centers):
        if not is_black(center):
            continue
        black_clusters_count += 1
        mask = (labels == i).astype("uint8") * 255
        mask = mask.reshape((height, width))

```

```

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, kernel_size)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel, iterations=iterations)
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel, iterations=iterations)

contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    area = cv2.contourArea(cnt)
    if area > min_cluster_size:
        x, y, w, h = cv2.boundingRect(cnt)
        aspect_ratio = max(w, h) / min(w, h) if min(w, h) > 0 else 1
        circularity = calculate_circularity(cnt)
        vertices = calculate_vertices(cnt)
        shape = determine_shape(vertices, circularity, aspect_ratio)
        if shape:
            centroid = calculate_centroid(cnt)
            obj_gray = image_gray[y:y+h, x:x+w]
            texture = extract_texture_features_lbp(obj_gray)
            clusters.append({
                'bbox': (x, y, w, h),
                'centroid': centroid,
                'area': area,
                'circularity': circularity,
                'vertices': vertices,
                'shape': shape,
                'cluster_id': i,
                'texture': texture
            })
            print(f"Пара {pair_id}: Обробка кластера {i}:")
            print(f" Площа: {area}")
            print(f" Співвідношення сторін: {aspect_ratio:.2f}")
            print(f" Круговість: {circularity:.2f}")
            print(f" Кількість вершин: {vertices}")
            print(f" Форма: {shape}")
print(f"Кількість чорних кластерів: {black_clusters_count}")
return clusters

def match_clusters(clusters_prev, clusters_curr, max_distance=100):
    if not clusters_prev or not clusters_curr:
        return []

    cost_matrix = np.zeros((len(clusters_prev), len(clusters_curr)), dtype=np.float32)
    for i, obj_prev in enumerate(clusters_prev):
        for j, obj_curr in enumerate(clusters_curr):
            dist_centroid = np.linalg.norm(np.array(obj_prev['centroid']) -
np.array(obj_curr['centroid']))

```

```

dist_texture = np.linalg.norm(obj_prev['texture'] - obj_curr['texture'])

texture_weight = 0.5
dist = dist_centroid + texture_weight * dist_texture

if obj_prev['shape'] != obj_curr['shape'] or abs(obj_prev['area'] - obj_curr['area']) > 0.3
* obj_prev['area']:
    dist += max_distance

cost_matrix[i, j] = dist

row_ind, col_ind = linear_sum_assignment(cost_matrix)

matched = []
for i, j in zip(row_ind, col_ind):
    if cost_matrix[i, j] < max_distance:
        movement = (
            clusters_curr[j]['centroid'][0] - clusters_prev[i]['centroid'][0],
            clusters_curr[j]['centroid'][1] - clusters_prev[i]['centroid'][1]
        )
        matched.append({
            'id': i + 1,
            'prev': clusters_prev[i],
            'current': clusters_curr[j],
            'movement': movement,
            'distance_px': np.linalg.norm(movement)
        })
return matched

def draw_movement(prev_image, curr_image, matched_objects, pixels_per_mm):
    prev_annotated = prev_image.copy()
    curr_annotated = curr_image.copy()

    for obj in matched_objects:
        obj_id = obj['id']
        prev = obj['prev']
        curr = obj['current']
        movement = obj['movement']
        shape = curr['shape']

        dx_px, dy_px = movement
        dx_mm = dx_px / pixels_per_mm if pixels_per_mm != 0 else 0
        dy_mm = dy_px / pixels_per_mm if pixels_per_mm != 0 else 0

        x_prev, y_prev, w_prev, h_prev = prev['bbox']

```

```

cv2.rectangle(prev_annotated, (x_prev, y_prev), (x_prev + w_prev, y_prev + h_prev), (255, 0,
0), 2)
cv2.putText(prev_annotated, f"Obj {obj_id} ({shape})", (x_prev, y_prev - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)

x_curr, y_curr, w_curr, h_curr = curr['bbox']
cv2.rectangle(curr_annotated, (x_curr, y_curr), (x_curr + w_curr, y_curr + h_curr), (0, 255,
0), 2)
cv2.putText(curr_annotated, f"Obj {obj_id} ({shape})", (x_curr, y_curr - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

start_point = (int(prev['centroid'][0]), int(prev['centroid'][1]))
end_point = (int(curr['centroid'][0]), int(curr['centroid'][1]))
cv2.arrowedLine(curr_annotated, start_point, end_point, (0, 0, 255), 2, tipLength=0.3)

movement_text = f"dx={dx_mm:.2f} mm, dy={dy_mm:.2f} mm"
cv2.putText(curr_annotated, movement_text, (end_point[0] + 10, end_point[1] + 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)

return prev_annotated, curr_annotated

```

```
class Application:
```

```
def __init__(self, master):
```

```
    global app
```

```
    self.master = master
```

```
    app = self
```

```
    master.title("Аналіз Зміщення Об'єктів (K-Means)")
```

```
    master.geometry("1600x1000")
```

```
    master.resizable(True, True)
```

```
    self.frame_top = Frame(master)
```

```
    self.frame_top.pack(pady=10, fill='x')
```

```
    self.button_clear = Button(
```

```
        self.frame_top, text="Очистити", command=self.clear_results, width=15
```

```
)
```

```
    self.button_clear.pack(side=LEFT, padx=10)
```

```
    self.button_quit = Button(
```

```
        self.frame_top, text="Вийти", command=self.quit_program, width=15
```

```
)
```

```
    self.button_quit.pack(side=LEFT, padx=10)
```

```
    self.button_process = Button(
```

```
        self.frame_top, text="Запустити Аналіз", command=self.run_processing_two_images, width=20
```

```

)
self.button_process.pack(side=LEFT, padx=10)

self.label_title = Label(master, text="Аналіз Зміщення Об'єктів (K-Means)", font=("Helvetica",
16))
self.label_title.pack(pady=10)

self.frame_actions = Frame(master)
self.frame_actions.pack(pady=10, fill='x')

self.button_browse_image1 = Button(
    self.frame_actions, text="Вибрати Перше Зображення", command=self.browse_image1, width=30
)
self.button_browse_image1.pack(side=LEFT, padx=10, pady=5)

self.label_image1 = Label(self.frame_actions, text="Перше зображення не вибрано.")
self.label_image1.pack(side=LEFT, padx=10, pady=5)

self.button_browse_image2 = Button(
    self.frame_actions, text="Вибрати Друге Зображення", command=self.browse_image2, width=30
)
self.button_browse_image2.pack(side=LEFT, padx=10, pady=5)

self.label_image2 = Label(self.frame_actions, text="Друге зображення не вибрано.")
self.label_image2.pack(side=LEFT, padx=10, pady=5)

self.frame_settings = Frame(master)
self.frame_settings.pack(pady=10, fill='x')

self.label_k = Label(self.frame_settings, text="Кількість Кластерів (k):", font=("Helvetica",
12))
self.label_k.grid(row=0, column=0, padx=10, pady=5, sticky='e')

self.slider_k = Scale(self.frame_settings, from_=2, to=20, orient=HORIZONTAL)
self.slider_k.set(5)
self.slider_k.grid(row=0, column=1, padx=10, pady=5)

self.label_min_cluster_size = Label(self.frame_settings, text="Мінімальна площа кластера:",
font=("Helvetica", 12))
self.label_min_cluster_size.grid(row=0, column=2, padx=10, pady=5, sticky='e')

self.slider_min_cluster_size = Scale(self.frame_settings, from_=1000, to=5000,
orient=HORIZONTAL)
self.slider_min_cluster_size.set(1000)
self.slider_min_cluster_size.grid(row=0, column=3, padx=10, pady=5)

```

```

self.label_num_calibration_objects = Label(self.frame_settings, text="Кількість об'єктів для
калібрування:", font=("Helvetica", 12))
self.label_num_calibration_objects.grid(row=0, column=4, padx=10, pady=5, sticky='e')

self.slider_num_calibration_objects = Scale(self.frame_settings, from_=1, to=10,
orient=HORIZONTAL)
self.slider_num_calibration_objects.set(5)
self.slider_num_calibration_objects.grid(row=0, column=5, padx=10, pady=5)

self.entry_scale = Entry(self.frame_settings, width=10, state='readonly')

self.entry_known_shift = Entry(self.frame_settings, width=10)
self.entry_known_shift.insert(0, str(KNOWN_SHIFT_MM_DEFAULT))

self.var_known_shift = IntVar()
self.var_known_shift.set(1)

self.progress = Progressbar(master, orient="horizontal", length=1200, mode="determinate")
self.progress.pack(pady=10)

self.status = Label(master, text="Готово", font=("Helvetica", 12))
self.status.pack(pady=5)

self.frame_images = Frame(master)
self.frame_images.pack(pady=10, fill='x')

self.label_prev_title = Label(self.frame_images, text="Анотоване Перше Зображення:",
font=("Helvetica", 12))
self.label_prev_title.pack(side=LEFT, padx=10, pady=5)
self.label_prev_image = Label(self.frame_images)
self.label_prev_image.pack(side=LEFT, padx=10, pady=5, fill='both', expand=True)

self.label_curr_title = Label(self.frame_images, text="Анотоване Друге Зображення:",
font=("Helvetica", 12))
self.label_curr_title.pack(side=LEFT, padx=10, pady=5)
self.label_curr_image = Label(self.frame_images)
self.label_curr_image.pack(side=LEFT, padx=10, pady=5, fill='both', expand=True)

self.frame_results = Frame(master)
self.frame_results.pack(pady=10, fill='both', expand=True)

self.label_list = Label(self.frame_results, text="Список зміщених об'єктів:",
font=("Helvetica", 12))
self.label_list.pack(pady=5)

self.frame_text = Frame(self.frame_results)

```

```

self.frame_text.pack(pady=5, fill='both', expand=True)

self.scrollbar = Scrollbar(self.frame_text)
self.scrollbar.pack(side=RIGHT, fill=Y)

self.listbox = Text(self.frame_text, width=150, height=20, yscrollcommand=self.scrollbar.set)
self.listbox.pack(side=LEFT, fill="both", expand=True)
self.scrollbar.config(command=self.listbox.yview)

self.result_queue = queue.Queue()

self.toggle_known_shift()

self.dpi_x = None
self.dpi_y = None
self.pixels_per_mm = None

def toggle_known_shift(self):
    if self.var_known_shift.get():
        self.entry_known_shift.config(state='normal')
    else:
        self.entry_known_shift.config(state='disabled')
        self.entry_known_shift.delete(0, END)
        self.entry_known_shift.insert(0, "")

def browse_image1(self):
    image_path = filedialog.askopenfilename(
        filetypes=[("Image Files", "*.jpg;*.jpeg;*.png;*.bmp;*.tiff")]
    )
    if image_path:
        try:
            pil_image = Image.open(image_path)
            dpi = pil_image.info.get('dpi', (72, 72))
            self.dpi_x, self.dpi_y = dpi
            print(f"Зчитано DPI першого зображення: {self.dpi_x}, {self.dpi_y}")
        except Exception as e:
            messagebox.showerror("Помилка", f"Не вдалося прочитати зображення: {e}")
            return
        self.image1_path = image_path
        self.label_image1.config(text=f"Перше зображення: {os.path.basename(image_path)}")

    if self.dpi_x and self.dpi_y:
        if self.dpi_x != self.dpi_y:
            messagebox.showwarning("Попередження", "Горизонтальний та вертикальний DPI різні. Використовується середнє значення.")
            average_dpi = (self.dpi_x + self.dpi_y) / 2

```

```

        self.pixels_per_mm = average_dpi / 25.4
        print(f"Розраховано pixels_per_mm: {self.pixels_per_mm:.4f}")
        self.entry_scale.config(state='normal')
        self.entry_scale.delete(0, END)
        self.entry_scale.insert(0, f"{self.pixels_per_mm:.4f}")
        self.entry_scale.config(state='readonly')
    else:
        self.pixels_per_mm = None
        self.entry_scale.config(state='normal')
        self.entry_scale.delete(0, END)
        self.entry_scale.insert(0, "1.0")
        self.pixels_per_mm = 1.0
        messagebox.showinfo("Інформація", "DPI не знайдено. Масштаб встановлено на 1.0 рх/мм.")

def browse_image2(self):
    image_path = filedialog.askopenfilename(
        filetypes=[("Image Files", "*.jpg;*.jpeg;*.png;*.bmp;*.tiff")]
    )
    if image_path:
        try:
            pil_image = Image.open(image_path)
            dpi = pil_image.info.get('dpi', (72, 72))
            self.dpi_x2, self.dpi_y2 = dpi
            print(f"Зчитано DPI другого зображення: {self.dpi_x2}, {self.dpi_y2}")
        except Exception as e:
            messagebox.showerror("Помилка", f"Не вдалося прочитати зображення: {e}")
            return
        self.image2_path = image_path
        self.label_image2.config(text=f"Друге зображення: {os.path.basename(image_path)}")

    if self.dpi_x2 and self.dpi_y2:
        if self.dpi_x2 != self.dpi_y2:
            messagebox.showwarning("Попередження", "Горизонтальний та вертикальний DPI різні. Використовується середнє значення.")
            average_dpi = (self.dpi_x2 + self.dpi_y2) / 2
            pixels_per_mm_image2 = average_dpi / 25.4
            print(f"Розраховано pixels_per_mm другого зображення: {pixels_per_mm_image2:.4f}")

        if self.pixels_per_mm:
            if not np.isclose(self.pixels_per_mm, pixels_per_mm_image2, atol=0.01):
                messagebox.showwarning("Попередження", "Масштаб першого та другого зображень різний. Використовується середнє значення.")
                self.pixels_per_mm = (self.pixels_per_mm + pixels_per_mm_image2) / 2
                print(f"Новий pixels_per_mm після усереднення: {self.pixels_per_mm:.4f}")
                self.entry_scale.config(state='normal')
                self.entry_scale.delete(0, END)

```

```

        self.entry_scale.insert(0, f"{self.pixels_per_mm:.4f}")
        self.entry_scale.config(state='readonly')
    else:
        self.pixels_per_mm = pixels_per_mm_image2
        print(f"Позраховано pixels_per_mm: {self.pixels_per_mm:.4f}")
        self.entry_scale.config(state='normal')
        self.entry_scale.delete(0, END)
        self.entry_scale.insert(0, f"{self.pixels_per_mm:.4f}")
        self.entry_scale.config(state='readonly')
    else:
        if not self.pixels_per_mm:
            self.pixels_per_mm = 1.0
            self.entry_scale.config(state='normal')
            self.entry_scale.delete(0, END)
            self.entry_scale.insert(0, "1.0")
            messagebox.showinfo("Інформація", "DPI не знайдено для другого зображення. Масштаб
встановлено на 1.0 px/mm.")

def run_processing_two_images(self):
    if not hasattr(self, 'image1_path') or not hasattr(self, 'image2_path'):
        messagebox.showwarning("Попередження", "Будь ласка, виберіть обидва зображення для
аналізу.")
    return
    k = self.slider_k.get()
    min_cluster_size = self.slider_min_cluster_size.get()

    known_shift = self.var_known_shift.get() == 1
    if known_shift:
        try:
            known_shift_mm = float(self.entry_known_shift.get())
            if known_shift_mm <= 0:
                raise ValueError
        except ValueError:
            messagebox.showerror("Помилка", "Відоме зміщення має бути позитивним числом.")
            return
    else:
        known_shift_mm = None

    result_folder = os.path.join(os.path.dirname(self.image1_path), RESULTS_FOLDER)
    if not os.path.exists(result_folder):
        os.makedirs(result_folder)

    for filename in os.listdir(result_folder):
        if filename.startswith('mask_') or filename.startswith('annotated_') or
filename.endswith('.csv') or filename.startswith('segmented'):
            os.remove(os.path.join(result_folder, filename))

```

```

self.progress['value'] = 0
self.status.config(text="Обработка...")

processing_thread = threading.Thread(
    target=process_image_pair,
    args=(
        self.image1_path,
        self.image2_path,
        self.progress,
        self.status,
        self.listbox,
        self.result_queue,
        k,
        min_cluster_size,
        1,
        result_folder,
        known_shift_mm,
        self,
        50
    ),
    daemon=True
)
processing_thread.start()
self.master.after(100, self.check_result_queue)

def check_result_queue(self):
    try:
        while True:
            message = self.result_queue.get_nowait()
            if message[0] == "done":
                self.status.config(text=message[1])
            elif message[0] == "error":
                self.status.config(text=message[1])
                messagebox.showerror("Помилка", message[1])
    except queue.Empty:
        pass
    self.master.after(100, self.check_result_queue)

def clear_results(self):
    self.listbox.delete('1.0', END)
    self.listbox.insert(END, "Obj ID\t Shape\t Total Displacement (px)\t Total Displacement
(mm)\n")
    self.listbox.insert(END, "-----\n")
    self.status.config(text="Очищено")
    self.progress['value'] = 0

```

```

self.label_prev_image.config(image='')
self.label_prev_image.image = None
self.label_curr_image.config(image='')
self.label_curr_image.image = None
self.entry_scale.config(state='normal')
self.entry_scale.delete(0, END)
self.entry_scale.insert(0, "")
self.entry_scale.config(state='readonly')

def quit_program(self):
    self.master.quit()

def process_image_pair(image1_path, image2_path, progress_bar, status_label, listbox, result_queue, k,
min_cluster_size, pair_id, result_folder_path, known_shift_mm, app, max_distance=50):
    image1 = cv2.imread(image1_path)
    image2 = cv2.imread(image2_path)
    if image1 is None or image2 is None:
        result_queue.put(("error", f"Неможливо відкрити зображення: {image1_path}, {image2_path}"))
        return

    image1_preprocessed, image1_resized, thresh1 = preprocess_image(image1)
    image2_preprocessed, image2_resized, thresh2 = preprocess_image(image2)

    segmented1, labels1, centers1 = kmeans_clustering(image1_preprocessed, k=k)
    segmented2, labels2, centers2 = kmeans_clustering(image2_preprocessed, k=k)

    segmented1_path = os.path.join(result_folder_path, f'segmented1_pair_{pair_id}.jpg')
    segmented2_path = os.path.join(result_folder_path, f'segmented2_pair_{pair_id}.jpg')
    cv2.imwrite(segmented1_path, segmented1)
    cv2.imwrite(segmented2_path, segmented2)
    print(f"Сегментовані зображення збережено: {segmented1_path}, {segmented2_path}")

    clusters1 = extract_clusters(segmented1, labels1, centers1, image1_preprocessed[:, :, 0],
                                min_cluster_size=min_cluster_size, pair_id=pair_id)
    clusters2 = extract_clusters(segmented2, labels2, centers2, image2_preprocessed[:, :, 0],
                                min_cluster_size=min_cluster_size, pair_id=pair_id)

    print(f"Пара {pair_id}: Знайдено {len(clusters1)} об'єктів у першому зображенні.")
    print(f"Пара {pair_id}: Знайдено {len(clusters2)} об'єктів у другому зображенні.")

    if not clusters1 or not clusters2:
        result_queue.put(("error", f"Пара {pair_id}: Не вдалося знайти об'єкти у одному з зображень."))
        return

    matched_objects = match_clusters(clusters1, clusters2, max_distance=max_distance)

```

```

if not matched_objects:
    result_queue.put(("error", f"Пара {pair_id}: Не знайдено збігів між об'єктами."))
    return

if known_shift_mm and app.pixels_per_mm:
    calibration_movements = [obj['distance_px'] for obj in
matched_objects[:app.slider_num_calibration_objects.get()]]
    if calibration_movements:
        average_movement_px = np.mean(calibration_movements)
        pixels_per_mm_calibrated = average_movement_px / known_shift_mm
        app.pixels_per_mm = pixels_per_mm_calibrated
        print(f"Калібрування: pixels_per_mm = {app.pixels_per_mm:.4f}")

        app.entry_scale.config(state='normal')
        app.entry_scale.delete(0, END)
        app.entry_scale.insert(0, f"{app.pixels_per_mm:.4f}")
        app.entry_scale.config(state='readonly')

        for obj in matched_objects:
            distance_px = obj['distance_px']
            distance_mm = distance_px / app.pixels_per_mm if app.pixels_per_mm != 0 else 0
            obj['distance_mm'] = distance_mm
    else:
        result_queue.put(("error", f"Пара {pair_id}: Не вдалося здійснити калібрування масштабу."))
        return
else:
    if app.pixels_per_mm:
        pixels_per_mm = app.pixels_per_mm
        print(f"Використовується pixels_per_mm: {pixels_per_mm:.4f}")
    else:
        pixels_per_mm = 1.0
        print("Масштаб не визначено. Встановлено pixels_per_mm на 1.0 px/mm.")
    for obj in matched_objects:
        distance_px = obj['distance_px']
        distance_mm = distance_px / pixels_per_mm
        obj['distance_mm'] = distance_mm

prev_annotated, curr_annotated = draw_movement(image1_resized, image2_resized, matched_objects,
app.pixels_per_mm if app.pixels_per_mm else 1.0)
prev_annotated_path = os.path.join(result_folder_path, f'annotated_prev_pair_{pair_id}.jpg')
curr_annotated_path = os.path.join(result_folder_path, f'annotated_current_pair_{pair_id}.jpg')
cv2.imwrite(prev_annotated_path, prev_annotated)
cv2.imwrite(curr_annotated_path, curr_annotated)
print(f"Анотовані зображення збережено: {prev_annotated_path}, {curr_annotated_path}")

prev_annotated_rgb = cv2.cvtColor(prev_annotated, cv2.COLOR_BGR2RGB)

```

```

curr_annotated_rgb = cv2.cvtColor(curr_annotated, cv2.COLOR_BGR2RGB)

prev_pil = Image.fromarray(prev_annotated_rgb)
curr_pil = Image.fromarray(curr_annotated_rgb)

max_display_size = (700, 600)
prev_pil.thumbnail(max_display_size, Image.Resampling.LANCZOS)
curr_pil.thumbnail(max_display_size, Image.Resampling.LANCZOS)

prev_photo = ImageTk.PhotoImage(prev_pil)
curr_photo = ImageTk.PhotoImage(curr_pil)

app.label_prev_image.config(image=prev_photo)
app.label_prev_image.image = prev_photo
app.label_curr_image.config(image=curr_photo)
app.label_curr_image.image = curr_photo

listbox.insert(END, f"Папа {pair_id}\n")
listbox.insert(END, "Obj ID\t Shape\t Total Displacement (mm)\n")
listbox.insert(END, "-----\n")

print(END, "-----\n")
for obj in matched_objects:
    obj_id = obj['id']
    shape = obj['current']['shape']
    distance_px = obj['distance_px']
    distance_mm = obj['distance_mm']
    listbox.insert(END, f"{obj_id}\t {shape}\t\t\t {distance_mm:.2f}\n")
    print(END, f"{obj_id}\t {shape}\t\t\t {distance_mm:.2f}\n")

listbox.insert(END, "\n")

progress_bar['value'] = 100
status_label.config(text=f"Папа {pair_id}: Обработка завершена.")
result_queue.put(("done", f"Папа {pair_id}: Обработка завершена.))

def main():
    root = Tk()
    app = Application(root)
    root.mainloop()

if __name__ == "__main__":
    main()

```

Додаток Б

Технічні науки

ISSN 2307-5732

DOI 10.31891/2307-5732-2024-339-4-10
УДК 004+627.328.9

ВУСАТИЙ НІКИТА

Хмельницький національний університет
<https://orcid.org/0009-0005-1518-7087>
e-mail: mixgguo@gmail.com

ПАСІЧНИК ОЛЕКСАНДР

Хмельницький національний університет
<https://orcid.org/0000-0002-8760-4688>
e-mail: o.pasichnyk@gmail.com

МАНЗІУК ЕДУАРД

Хмельницький національний університет
<https://orcid.org/0000-0002-7310-2126>
e-mail: eduard.em.km@gmail.com

СКРИПНИК ТЕТЯНА

Хмельницький національний університет
<https://orcid.org/0000-0002-8531-5348>
e-mail: tskrypnik1970@gmail.com**СПОСІБ ОЦІНКИ ЗМІЩЕННЯ ОБ'ЄКТІВ ЗА АНАЛІЗОМ ЗОБРАЖЕНЬ З КРУГОВИМ ГРАДІЄНТОМ КОЛЬОРУ У ЗАДАЧІ ВИМІРЮВАННЯ В ТЕХНІЧНИХ СИСТЕМАХ**

Нинішній етап розвитку людства можна характеризувати як неймовірно великий обсяг інформації у цифровій індустрії. Спостерігаючи за цим, сміливо можна сказати, цифрова індустрія - повністю замінила та автоматизувало життя і життєдіяльність. Як було зазначено, сьогодні все більш поширюється тенденція до повного проникнення інтелектуальних інформаційних технологій у всі сфери людської діяльності, що часто супроводжується використанням різних цифрових технічних пристроїв. Однією зі стійких тенденцій останнього часу є значне поширення використання засобів для отримання зображення навіть у повсякденному житті. У той же час, зростає потреба в визначенні наявності або відсутності зміщення об'єктів спостереження. Ці обставини створюють можливості для розв'язання цієї задачі шляхом порівняння послідовних зображень, що дозволяє не тільки якісно оцінювати наявність зміщення, але й вимірювати його величину. Такий підхід надає можливість вирішувати задачі вимірювання в технічних системах. Ключовим питанням, яке обумовлює потенційну точність в задачі вимірювання, є максимально точне вимірювання положення об'єкту спостереження на тлі інших. Загалом спостерігач має справу із градієнтною зміною кольорів в умовах як природного так й штучного освітлення, й завдання визначення положення об'єкту спостереження фактично полягає в аналізі зображень з круговим градієнтом кольору.

Ключові слова: спосіб зміщення, аналіз зображень, метод k-means.

VUSATYI NIKITA, PASICHNYK OLEKSANDR, MANZIUK EDUARD, SKRYPNYK TETIANA
Khmelnitskyi National University

A METHOD FOR ESTIMATING THE DISPLACEMENT OF OBJECTS BY ANALYZING IMAGES WITH A CIRCULAR COLOR GRADIENT IN THE PROBLEM OF MEASUREMENT IN TECHNICAL SYSTEMS

The current stage of human development can be characterized as an incredibly large amount of information in the digital industry. Observing this, we can safely say that the digital industry has completely replaced and automated life and activities. As noted above, today there is a growing trend towards the full penetration of intelligent information technologies into all spheres of human activity, often accompanied by the use of various digital technical devices. One of the recent stable trends is the widespread use of imaging devices even in everyday life. At the same time, there is a growing need to determine the presence or absence of displacement of objects of observation. These circumstances create opportunities for solving this problem by comparing successive images, which allows not only to qualitatively assess the presence of displacement, but also to measure its magnitude. This approach makes it possible to solve measurement problems in technical systems. The key issue that determines the potential accuracy in a measurement task is the most accurate measurement of the position of the object of observation against the background of others. In general, the observer deals with a gradient change in color under both natural and artificial lighting conditions, and the task of determining the position of the object of observation is actually to analyze images with a circular color gradient.

Keywords: displacement method, image analysis, k-means method.

Постановка проблеми

Серед різноманітних видів інформації, що повсякденно використовуються людиною у своїй науковій, виробничій та побутовій діяльності, першочергове значення має вимірювальна інформація, яка містить кількісну оцінку результатів наукових досліджень і спостережень, стан виробництва та технологічні процеси, характеристики продукції, фактори зовнішнього середовища, фізіологічні та психоемоційні стани тощо [1–5].

Загальносвітова тенденція вдосконалення обладнання та технологій полягає у покращенні майже кожного продукту. Фактором, який об'єктивно впливає на підвищення їх якості, надійності та довговічності, є дослідження відповідних властивостей, параметрів і характеристик матеріалів, деталей, вузлів, агрегатів і систем з деталізованим рівнем інформації щодо їх збереження та накопичення, їх структурування, систематизація, класифікація тощо.

Вченим і дослідникам завжди доводилося з'ясовувати максимально повну і точну інформацію про процеси і явища, що відбуваються під час кожного експерименту. Проблема підвищення інформативності спостережень і достовірності отриманих експериментальних даних може бути вирішена за рахунок збільшення обсягу отриманих даних, зокрема за рахунок скорочення часу між дискретними вимірюваннями, збереження безперервного вимірювання параметрів процесу без зупинки і без розбирання, використання відповідних методів реєстрації, які не спотворюють спостережувані закономірності, використання різноманітних методів візуалізації, використання найновіших методів і технік вимірювання, які базуються на принципово інших фізичних процесах чи явищах.

Одним з перспективних підходів до вирішення цих завдань і проблем є впровадження різноманітних систем автоматичної реєстрації параметрів процесу.

Методи візуалізації та документування, які використовують фотографію, популярні в наукових дослідженнях протягом майже двох століть. Широке використання фотографії пояснюється здатністю документувати процеси та явища як якісно, так і кількісно. Широке розповсюдження засобів отримання цифрових зображень об'єктивно обумовлює розширення кола вирішуваних задач з їх використанням. Покращення якості та достовірності отримуваних результатів є можливим за умови використання методів та алгоритмів шугучного інтелекту та інтелектуального аналізу даних [6–14].

Аналіз останніх джерел

Розглянуті сучасні технології оцінки поля зміщення та сегментація зображень за допомогою блокового зіставлення, підсиленого нейронною мережею [15]. Розрахунки, присвячені оцінці зміщення в ультразвукових зображеннях з використанням псевдофаз [16].

Метою роботи є розробка способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Виклад основного матеріалу

Загальносвітова тенденція вдосконалення обладнання та технологій полягає у покращенні майже кожного продукту. Фактором, який об'єктивно впливає на підвищення їх якості, надійності та довговічності, є дослідження відповідних властивостей, параметрів і характеристик матеріалів, деталей, вузлів, агрегатів і систем з деталізованим рівнем інформації щодо їх збереження та накопичення, їх структурування, систематизація, класифікація тощо.

Вченим і дослідникам завжди доводилося з'ясовувати максимально повну і точну інформацію про процеси і явища, що відбуваються під час кожного експерименту. Проблема підвищення інформативності спостережень і достовірності отриманих експериментальних даних може бути вирішена за рахунок збільшення обсягу отриманих даних, зокрема за рахунок скорочення часу між дискретними вимірюваннями, збереження безперервного вимірювання параметрів процесу без зупинки і без розбирання, використання відповідних методів реєстрації, які не спотворюють спостережувані закономірності, використання різноманітних методів візуалізації, використання найновіших методів і технік вимірювання, які базуються на принципово інших фізичних процесах чи явищах [17].

Одним з перспективних підходів до вирішення цих завдань і проблем є впровадження різноманітних систем автоматичної реєстрації параметрів процесу.

Методи візуалізації та документування, які використовують фотографію, популярні в наукових дослідженнях протягом майже двох століть. Широке використання фотографії пояснюється здатністю документувати процеси та явища як якісно, так і кількісно [18].

Особливості фотографічних методів фіксації інформації в результаті досліджень і спостережень мають значну перевагу і тому вони так широко використовуються в багатьох сферах життя людини [19].

Однією з найпопулярніших тенденцій сучасного вимірювання є значна роль інформаційних технологій. Одним з життєздатних рішень проблеми вимірювання за допомогою цього підходу є непрямі методи, які базуються на обробці зображень.

Наприклад, науково-технічна мета підвищення ефективності методів вимірювання може бути реалізована шляхом оптико-електронного позиціонування та пасивних стереоскопічних вимірювань дальності завдяки розробці та впровадженню перешкодостійких методів обробки оптичного потоку.

У проекті параметри переміщень об'єктів розраховуються за алгоритмічною обробкою їх зображень, зокрема запропоновано два способи визначення розмірів геометричної натури - шляхом виділення контурів, що переходять у символічний опис зображення, або шляхом сегментації зображення за рівнем яскравості та визначенням координат точки на контурі об'єкта. У результаті прогресування цих зображень дозволяє ідентифікувати лінійні та кутові рухи [20].

Інша важлива сфера, яка потребує, з одного боку, результатів вимірювань, а з іншого - первинної інформації, а саме зображень, представлена в системах технічного зору.

У підсумку в статті описані галузі техніки і науки, які активно використовують різні системи технічного зору. У статті викладено основні підходи до використання таких систем, запропоновано нову формулу для розрахунку відстані між об'єктами та наведено її аналіз [21].

Реалізація вимірювальних систем, які передбачають використання зображень, потребує використання певних алгоритмів цифрової обробки зображень і комп'ютерного зору.

У результаті проект містить дослідницькі матеріали щодо створення структурних методів розпізнавання спостережуваних об'єктів на відповідних зображеннях у контексті геометричних перетворень

за наявності інтерференції. У цій статті описано сучасні підходи до створення структурного опису зображення, створення структур і асоціацій ознак, обчислення мір подібності між описами, використання методів обробки зображень для прискорення процесу або підвищення його точності при вирішенні практичних завдань, які включають аналіз зображення.

Дослідження сучасних наукових публікацій на тему вимірювання лінійних переміщень за допомогою аналізу зображень підтверджує можливість розробки багатьох методів вимірювання, зазначений підхід є ефективним і має високу ступінь застосовності.

Алгоритм кластеризації k-means (k-means clustering) – це техніка машинного навчання, яка використовується для кластеризації подібних об'єктів у класи на основі їхніх властивостей і характеристик. В алгоритмі k-середніх кожен кластер представлений своїм центроїдом, який є точкою, яка є середнім значенням об'єктів у кластері.

Мета кластеризації k-середніх полягає в тому, щоб розділити набір даних на підмножини (кластери), кожна з яких містить об'єкти, схожі один на одного, і об'єкти, які не схожі один на одного. Кластеризація k-середніх використовується в різних галузях, включаючи медицину, біологію, маркетинг тощо.

При використанні методу k-середніх для реалізації способу оцінки зміщення об'єктів за аналізом зображень кластером є окремий об'єкт спостереження на цьому зображенні. Застосування кластеризації дозволяє визначити положення та, в подальшому, зміщення одразу кількох об'єктів спостереження з не синхронізованим рухом.

Головною метою кластеризації k-середніх це пошук кластерів у заданих вхідних параметрах. Існує кілька способів цього досягти. Можна використовувати метод через випробування, та отримувати безліч помилок і саме на цих помилках вчитися, вказавши значення K, наприклад, як 5, 6 або 7. І по мірі просування постійно змінювати відповідний параметр до того моменту, поки не буде отримано найкращий результат, а саме, найкращі, так звані, кластери.

Наступний метод використовує техніку Elbow. Цей метод полягає у визначенні значення K. Тож, коли система отримує значення K, воно випадковим чином призначає відповідну кількість центроїдів та визначає відстань кожної точки даних один від одних центроїдів. Таким чином, у кожного центроїда є свої точки. Відстань між точками та цим одним центроїдом є мінімальною. В результаті, усі точки призначені конкретного та відповідному центроїду, і призначені з мінімальною відстанню. На цьому етапі, отримано K кількість початкових, так званих, кластерів.

Для новоутворених кластерів визначається нове центральне положення. Положення центру тяжіння зміщується порівняно з випадковим розподілом.

Знову ж таки, відстань кожної точки обчислюється від нової точки центроїда. Якщо необхідно, точки даних переміщуються на нові центроїди, а середнє положення або новий центроїд знову обчислюється.

Якщо центроїд зміщується, ітерація продовжується, що вказує на те, що конвергенції не відбулося. Однак, як тільки центроїд перестане дрейфувати (що означає, що процес кластеризації завершено), результат буде відображено.

Висновки

За результатами виконання кваліфікаційної роботи було розроблено спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Реалізовано спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору. Суть способу полягає в опрацюванні серій послідовних зображень із визначенням на кожному з них положення об'єкту. Для визначення положення об'єкту використовується метод кластеризації, а саме метод k-means. Застосування методу кластеризації дозволяє визначення положення одразу кількох об'єктів на одиничному зображенні. Це дозволяє визначати величину зміщення кількох об'єктів одночасно за один цикл обробки однієї серії послідовної зображень, тобто оцінювати зміщення кількох об'єктів спостереження з несинхронізованим рухом без застосування ітераційної по об'єктовій процедури.

Література

1. Пасічник О.А. Програмна система методу вимірювання лінійних переміщень за аналізом зображень. Актуальні проблеми комп'ютерних наук. Збірник наукових праць за матеріалами XIII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». Хмельницький: ХНУ, 2021. С. 385–389.
2. Вараниця О.А., Пасічник О.А. Застосування об'єктно-орієнтованого програмування для автоматизації вимірювань на основі аналізу зображень. Збірник наукових праць факультету прикладної математики та комп'ютерних технологій Хмельницького національного університету. Хмельницький: ХНУ, 2010. С. 87–90.
3. Пасічник О.А. Застосування кількісної цифрової фотографії при трибологічних дослідженнях. Тези доповідей VI-ї міжнародної конференції молодих науковців "Інформатика і механіка". 2008. С. 19–20.
4. Пасічник О.А. Фотографічний метод вимірювань при трибологічних дослідженнях. Вісник Хмельницького національного університету. 2008. № 2(108). С. 173–175.

5. Пасичник О.А. Деякі методологічні аспекти застосування кількісної цифрової фотографії в наукових дослідженнях. Матеріали II міжнародної науково-практичної конференції "Дні науки 2006". Том 30. Дніпропетровськ: Наука і освіта, 2006. С. 10–12.
6. Manziuk E. A., Barmak O. V., Krak Iu. V., Pasichnyk O. A., Radiuk P. M., Mazurets O. V. Semantic alignment of ontologies meaningful categories with the generalization of descriptive structures. *Problems in programming*. 2022. Vol. 3, No. 4. P. 355-363. DOI: <https://doi.org/10.15407/pp2022.03-04.355>
7. Barmak A. V., Krak Y. V., Manziuk E. A., Kasianiuk V. S. Information technology of separating hyperplanes synthesis for linear classifiers. *Journal of Automation and Information Sciences*, 2019. Vol. 51, No. 5. P. 54–64. URL: <https://doi.org/10.1615/JAutomatInfScien.v51.i5.50>.
8. Krak I., Barmak O., Manziuk E. Using visual analytics to develop human and machine-centric models: A review of approaches and proposed information technology. *Computational Intelligence*, 2022. Vol. 38, No. 3. P. 921–946. URL: <https://doi.org/10.1111/coin.12289>.
9. Krak Iu. V., Kasianiuk V. S., Kudin H. I., Barmak O. V., Manziuk E. A. Multivariate scaling of the characteristic features based on pseudo-inverse operations for recognition problems solving. *Pattern Recognition and Image Analysis*, 2020. Vol. 30, No. 2. Pp. 184–191. URL: <https://doi.org/10.1134/S1054661820020078>.
10. Manziuk E. A., Barmak A. V., Krak Y. V., Kasianiuk V. S. Definition of information core for documents classification. *Journal of Automation and Information Sciences*, 2018. Vol. 50, No. 4. P. 25–34. URL: <https://doi.org/10.1615/JAutomatInfScien.v50.i4.30>.
11. Manziuk E. A., Wójcik W., Barmak O. V., Krak I. V., Kulias A. I., Drabovska V. A., Puhach V. M., Sundetov S., Mussabekova A. Approach to creating an ensemble on a hierarchy of clusters using model decisions correlation. *Przegląd Elektrotechniczny*, 2020. Vol. 96, No. 9. P. 108–113. URL: <https://doi.org/10.15199/48.2020.09.23>.
12. Krak I., Barmak O., Manziuk E., Kudin H. Approach to piecewise-linear classification in a multi-dimensional space of features based on plane visualization. *Lecture Notes in Computational Intelligence and Decision Making*. 2020. Vol. 1020. P. 35–47. URL: https://doi.org/10.1007/978-3-030-26474-1_3.
13. Krak I., Barmak O., Manziuk E., Kulias A. Data classification based on the features reduction and piecewise linear separation. *Intelligent Computing and Optimization*. 2020. Vol. 1072. P. 282–289. URL: https://doi.org/10.1007/978-3-030-33585-4_28.
14. Barmak O., Krak I., Manziuk E., Lytvynenko V., Kalyta O. Classification technology based on hyperplanes for visual analytics with implementations for different subject areas. *CEUR-WS*. 2020. Vol. 2623. P. 96–106.
15. Displacement estimation in ultrasound images using pseudo-phase. https://www.researchgate.net/publication/224466523_Displacement_estimation_in_ultrasound_images_using_pseudo-phase.
16. Displacement field estimation and image segmentation using block matching enhanced by a neural network. <https://www.csie.ntu.edu.tw/~fuh/personal/DisplacementFieldEstimationandImageSegmentation.pdf>.
17. Big data will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus - as long as the right policies and enablers are in place. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation>.
18. The Necessity of Measurement in Everyday Life. <https://www.bmdshapi.com/the-necessity-of-measurement-in-everyday-life-fred-b-bryant-ph-d-loyola-university-chicago/>.
19. Measurement in metrology, psychology and social sciences: data generation traceability and numerical traceability as basic methodological principles applicable across sciences. <https://link.springer.com/article/10.1007/s11135-020-00970-2>.
20. Recent advances and applications of deep learning methods in materials science. <https://www.nature.com/articles/s41524-022-00734-6>.
21. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. <https://www.mdpi.com/2072-4292/3/6/1104>.

References

1. Pasichnyk O.A. Programna systema metodu vymiruvannya liniinykh peremishchen za analizom zobrazhen. Aktualni problemy kompiuternykh nauk. Zbiryk naukovykh prats za materialamy XIII vseukrainskoi naukovo-praktychnoi konferentsii «Aktualni problemy kompiuternykh nauk APKN-2021». Khmelnytskyi: KhNU, 2021. S. 385–389.
2. Varanytsia O.A., Pasichnyk O.A. Zastosuvannya ob'iektno-orientovanoho prohramuvannya dla avtomatyzatsii vymiruvann na osnovi analizu zobrazhen. Zbiryk naukovykh prats fakultetu prykladnoi matematyky ta kompiuternykh tekhnologii Khmelnytskoho natsionalnogo universytetu. Khmelnytskyi: KhNU, 2010. S. 87–90.
3. Pasichnyk O.A. Zastosuvannya kilksnoi tsyfrovoi fotografii pry trybolohichnykh doslidzhenniakh. Tezy dopovidei VI-yi mizhnarodnoi konferentsii molodykh naukovtsiv "Informatyka i mekhanika". 2008. S. 19–20.
4. Pasichnyk O.A. Fotohrafichnyi metod vymiruvann pry trybolohichnykh doslidzhenniakh. Visnyk Khmelnytskoho natsionalnogo universytetu. 2008. № 2(108). S. 173–175.
5. Pasichnyk O.A. Deiaki metodolohichni aspekty zastosuvannya kilksnoi tsyfrovoi fotografii v naukovykh doslidzhenniakh. Materialy II mizhnarodnoi naukovo-praktychnoi konferentsii "Dni nauky 2006". Tom 30. Dnipropetrovsk: Nauka i osvita, 2006. S. 10–12.
6. Manziuk E. A., Barmak O. V., Krak Iu. V., Pasichnyk O. A., Radiuk P. M., Mazurets O. V. Semantic alignment of ontologies meaningful categories with the generalization of descriptive structures. *Problems in programming*. 2022. Vol. 3, No. 4. P. 355-363. DOI: <https://doi.org/10.15407/pp2022.03-04.355>
7. Barmak A. V., Krak Y. V., Manziuk E. A., Kasianiuk V. S. Information technology of separating hyperplanes synthesis for linear

- classifiers. *Journal of Automation and Information Sciences*, 2019. Vol. 51, No. 5. P. 54–64. URL: <https://doi.org/10.1615/JAutomatInfScien.v51.i5.50>.
8. Krak I., Barmak O., Manziuk E. Using visual analytics to develop human and machine-centric models: A review of approaches and proposed information technology. *Computational Intelligence*, 2022. Vol. 38, No. 3. P. 921–946. URL: <https://doi.org/10.1111/coim.12289>.
9. Krak Iu. V., Kasianiuk V. S., Kudin H. I., Barmak O. V., Manziuk E. A. Multivariate scaling of the characteristic features based on pseudo-inverse operations for recognition problems solving. *Pattern Recognition and Image Analysis*, 2020. Vol. 30, No. 2. Pp. 184–191. URL: <https://doi.org/10.1134/S1054661820020078>.
10. Manziuk E. A., Barmak A. V., Krak Y. V., Kasianiuk V. S. Definition of information core for documents classification. *Journal of Automation and Information Sciences*, 2018. Vol. 50, No. 4. P. 25–34. URL: <https://doi.org/10.1615/JAutomatInfScien.v50.i4.30>.
11. Manziuk E. A., Wójcik W., Barmak O. V., Krak I. V., Kulas A. I., Drabovska V. A., Puhach V. M., Sundetov S., Mussabekova A. Approach to creating an ensemble on a hierarchy of clusters using model decisions correlation. *Przegląd Elektrotechniczny*, 2020. Vol. 96, No. 9. P. 108–113. URL: <https://doi.org/10.15199/48.2020.09.23>.
12. Krak I., Barmak O., Manziuk E., Kudin H. Approach to piecewise-linear classification in a multi-dimensional space of features based on plane visualization. *Lecture Notes in Computational Intelligence and Decision Making*, 2020. Vol. 1020. P. 35–47. URL: https://doi.org/10.1007/978-3-030-26474-1_3.
13. Krak I., Barmak O., Manziuk E., Kulas A. Data classification based on the features reduction and piecewise linear separation. *Intelligent Computing and Optimization*, 2020. Vol. 1072. P. 282–289. URL: https://doi.org/10.1007/978-3-030-33583-4_28.
14. Barmak O., Krak I., Manziuk E., Lytvynenko V., Kalyta O. Classification technology based on hyperplanes for visual analytics with implementations for different subject areas. *CEUR-WS*, 2020. Vol. 2623. P. 96–106.
15. Displacement estimation in ultrasound images using pseudo-phase. https://www.researchgate.net/publication/224466523_Displacement_estimation_in_ultrasound_images_using_pseudo-phase.
16. Displacement field estimation and image segmentation using block matching enhanced by a neural network. <https://www.csie.ntu.edu.tw/~fuhs/personal/DisplacementFieldEstimationandImageSegmentation.pdf>.
17. Big data will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus - as long as the right policies and enablers are in place. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation>.
18. The Necessity of Measurement in Everyday Life. <https://www.bmdshapi.com/the-necessity-of-measurement-in-everyday-life-fred-b-bryant-ph-d-loyola-university-chicago/>.
19. Measurement in metrology, psychology and social sciences: data generation traceability and numerical traceability as basic methodological principles applicable across sciences. <https://link.springer.com/article/10.1007/s11135-020-00970-2>.
20. Recent advances and applications of deep learning methods in materials science. <https://www.nature.com/articles/s41524-022-00734-6>.
21. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. <https://www.mdpi.com/2072-4292/3/6/1104>.

Додаток ВГ

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»

15-16 листопада 2024

Хмельницький 2024

УДК 004:37:001:62

Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». Хмельницький, 2024. 582с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів, друкуються в авторській редакції та наведені в алфавітному порядку прізвищ авторів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів. Відповідальність за якість та зміст публікацій несе автор.

Участь у конференції та складові всіх її етапів (розгляд праць, перевірка на плагіат, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: apki.khnu@gmail.com

© 2024 Хмельницький національний університет
© 2024 Кафедра комп'ютерних наук ХНУ

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2024*XVI Всеукраїнська науково-практична конференція*

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

Робочі мови конференції:

українська, англійська

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

СПИСОК ОРГАНІЗАЦІЙ,**ПРЕДСТАВНИКИ ЯКИХ БРАЛИ УЧАСТЬ У РОБОТІ****КОНФЕРЕНЦІЇ:**

Донбаська державна машинобудівна академія
Західноукраїнський національний університет
Національний технічний університет «Харківський політехнічний інститут»
Національний університет «Львівська політехніка»
Приватний заклад вищої освіти «ІТ СТЕП Університет»
Сумський державний університет
Харківський національний університет радіоелектроніки
Хмельницький національний університет
Хмельницький фаховий економіко-технологічний коледж УЕП

ОРГКОМІТЕТ КОНФЕРЕНЦІЇ:

Олег СИНЮК – голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор.

Тетяна ГОВОРУЩЕНКО – заступник голови оргкомітету, декан факультету інформаційних технологій Хмельницького національного університету, доктор технічних наук, професор.

Олександр БАРМАК – заступник голови оргкомітету, завідувач кафедри комп'ютерних наук Хмельницького національного університету, доктор технічних наук, професор.

Олег САВЕНКО – професор кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету, доктор технічних наук, професор

Олена ВИСОЦЬКА – доктор технічних наук, завідувач кафедри радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. С. Жуковського «Харківський авіаційний інститут», професор

Євгеній ЛАВРОВ – доктор технічних наук, професор (Сумський державний університет)

Людмила ТИМОФЄЄВА – відповідальна за студентську науково-дослідну роботу ХНУ

Олександр МАЗУРЕЦЬ – секретар конференції, доцент кафедри комп'ютерних наук Хмельницького національного університету, к.т.н., доцент кафедри комп'ютерних наук ХНУ

Марина МОЛЧАНОВА – секретар конференції, викладач кафедри комп'ютерних наук Хмельницького національного університету

КОНТАКТНА ІНФОРМАЦІЯ:

e-mail для листування: apkt.khnu@gmail.com

Вусатий Н.О., Пасічник О.А., Скрипник Т.К.	
Сегментація зображень.....	112
Галилуйко С.В., Бабала Л.В.	
Проектування системи виявлення аномалій у мережевому трафіку на основі штучного інтелекту.....	115
Галицький О.С., Денисюк Д.О., Кожем'яко Я.Р.	
Стабілізація FPV-дрону за автоматично визначеною ціллю та подальше її спостереження.....	118
Гардиш Д.О., Мазурець О.В.	
Метод автоматизованого оцінювання відповідності тестових завдань семантичній складовій навчальних матеріалів за інтелектуальним аналізом їх текстового контенту.....	121
Гаркавюк Д.С., Петровський С.С., Вознюк Л.О.	
Метод виявлення пухлин мозку на зображеннях МРТ нейромережевими засобами.....	140
Гладун О.В., Мазурець О.В., Залуцька О.О.	
Аналіз прикладної імплементації методу нейромережевого виявлення емоційного стану людини у режимі реального часу.....	142
Говоруценко Т.О., Павлова О.О., Алексейко В.О., Кузьмін А.А.	
Аналіз потенційних загроз мобільних застосунків на основі дозволів з використанням методів машинного навчання.....	149
Голуб Ю.В., Базрій Р.О., Вознюк Л.О.	
Визначення кількості людей в натовпі у відеопотоці нейромережевими засобами ..	152
Григор'єв Д.М., Антипенко В.П.	
Алгоритм імплементації процесів логування та моніторингу в хмарних системах ..	156
Гуральник О.Б.	
Методи та засоби оброблення критичної інформації в корпоративних мережах ...	160
Гуцал Б.В.	
Кіберфізична система моніторингу бджолиних вуликів на основі IoT технологій ...	164
Денисенко Б.О., Молчанова М.О., Мазурець О.В.	
Інтелектуальна система виявлення дезінформації з застосуванням штучних нейронних мереж	167

УДК 004.93

Вусатий Н.О., Пасічник О.А., Скрипник Т.К.

*Хмельницький національний університет***СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ**

Розглянуто прикладні аспекти розробки методів сегментації зображень для точного виділення об'єктів на зображеннях в умовах наявності шуму. Запропонований підхід забезпечує високу точність сегментації для різноманітних типів зображень.

The paper considers applied aspects of developing image segmentation methods for accurate object detection in images in the presence of noise. The proposed approach provides high segmentation accuracy for various types of images.

Сегментація є основним етапом будь-якої подальшої обробки зображення в задачах аналізу чи розпізнавання. Проблема сегментації полягає в групуванні пікселів зображення в окремі регіони або об'єкти, спільні властивості яких дозволять вибрати елементи для подальших операцій. Наприклад, у сфері медичної діагностики сегментація може бути застосована для пошуку органів або пухлин на знімках, в системах безпеки - для автоматичного розпізнавання осіб. Майже всі алгоритми комп'ютерного зору працюють із високою точністю сегментації, але це стає тривіально важко через наявність шуму, нерівномірного освітлення та насиченого фону.

На сьогоднішній день розроблено багато методів сегментації зображень. Можна згадати наступні методи: Методи порогового значення, методи кластеризації та методи глибокого навчання.

Ще одне цікаве застереження для підходів на основі порогових значень полягає в тому, що оператор повинен встановити поріг яскравості, який буде відокремлювати об'єкти від фону. Дуже простий, досить ефективний, може не підійти у випадку зображень з нерівномірним освітленням, або складних фактурних зображень. Як правило, у методах кластеризації К-середніх і нечітких С-середніх пікселі групуються на основі подібності кольору або текстури. Як наслідок, можна отримати частини об'єктів зі слабкими, нечіткими межами — недоліки яких виявляються перевагами цих методів. З перевизначенням контрасту яскравості в обробці зображень продукти стають дедалі ефективнішими. Чим вони кращі, тим більше буде!

Деякі з найсучасніших методів сегментації зображень на основі глибокого навчання за допомогою архітектури нейронної мережі включають U-Net або Mask

R-CNN. Їх продуктивність забезпечує високий рівень точності навіть під час обробки складних зображень із неоднорідним фоном або високим рівнем шуму. Суть їх роботи заснована на попередньо навчених мережах для сегментації різних об'єктів на зображенні. Це дає підстави стверджувати, що ці методи є найбільш перспективними в проблемі сегментації зображення.

Проте, не зважаючи на цей великий прогрес, на достатньо гарний роботу усіх цих методів, вони мають свої недоліки, і вони серйозні. Серед недоліків є такий, що ці методи не підходять для обробки складних зображень. І навіть якщо взяти методи, які засновані на роботі глибокого навчання, вони все-одно вимагають значних обчислювальних ресурсів та потужної техніки. З чого можна зробити висновок у потребі нових підходів, які б об'єднували ці методи та забезпечували б надійну точність та якість.

Метою цього дослідження є об'єднання методів кластеризації з потужністю глибоких нейронних мереж у комплексний підхід до сегментації зображення, який забезпечує високу точність і адаптивність у шумному середовищі та за наявності гетерогенних елементів фону. Цей новий метод було розроблено з кінцевою метою досягнення балансу між точністю та ефективністю, щоб його можна було застосовувати.

Запропонований метод сегментації складається з чотирьох наступних етапів:

1. Попередня обробка зображення.
2. Кластеризація пікселів.
3. Глибока нейронна мережа для обробки кластерів.
4. Пост обробка контурів та усунення артефактів.

На першому етапі попередньої обробки застосовують кілька методів шумозаглушення. Один з них згладжує зображення за допомогою фільтрації Гауса і допомагає усунути незначні дефекти зображення. Після чого отримується краща якість сегментації та продуктивність кластеризації. Відповідно, він найкраще працює у застосуванні зображень, які мають високий рівень шуму. Ще одна найкраща практика для попередньої обробки - нормалізація кольорних каналів для рівномірного освітлення.

Другий етап - кластеризація інформації про колір і пікселів текстури. Процес ієрархічної сегментації зображення відбувається знизу вгору в тому сенсі, що пікселі об'єднуються відповідно до подібності, що призводить до дендрограмного представлення процесу об'єднання. Звичайно, деякі хибні групування присутні через вплив шуму або невеликі варіації значень рівня сірого, що належать до однієї області. Наприклад, у медичних зображеннях на основі

пікселів результати сегментації можна трансформувати в різні кластери однорідних областей, що представляють тканини або органи.

На третьому етапі реалізується глибока нейронна мережа на основі архітектури U-Net. Така архітектура об'єднує локальні та глобальні іміджеві функції. При застосуванні мережі контури об'єктів у кожному кластері чіткі. Нейронна мережа з багатьма шарами зможе враховувати абсолютно всі контекстні особливості об'єктів, що різко підвищує точність сегментації.

На останньому четвертому етапі, після завершення сегментації, може знадобитися видалити деякі артефакти, щоб зробити контури чіткішими. Як морфологічні операції застосовуються ерозія, при якій видаляються непотрібні деталі, і розширення, що покращує різкість країв об'єкта. Така пост обробка дозволяє підвищити якість кінцевого результату.

Запропонований підхід показав високу точність сегментації для складних зображень з неоднорідним фоном, зокрема при обробці медичних і супутникових знімків - значне зниження кількості помилкових спрацювань. Відповідно до результатів, які були отримані за допомогою різних методик, застосування кластеризації з глибокими нейронними мережами дозволяє отримувати набагато точніші результати. Справа в тому, що інші способи мають на увазі ймовірність невірної результату.

Подальші шляхи дослідження полягають в оптимізації розробленого методу для використання на мобільних пристроях і в хмарних сервісах. Важливим напрямком також є інтеграція розробленого підходу в системи автоматичного аналізу та інтерпретації зображень. Що може бути корисним у медичних дослідженнях, системах географічного моніторингу та безпеки.

Перелік посилань

1. Zhang, Y., Li, X. "Recent Advances in Image Segmentation: Techniques, Trends, and Applications." IEEE Transactions on Image Processing, 2021. URL: <https://ieeexplore.ieee.org/document/9195743>
2. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D. "Image Segmentation Using Deep Learning: A Survey." IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. URL: <https://ieeexplore.ieee.org/document/9271694>
3. He, K., Gkioxari, G., Dollár, P., Girshick, R. "Masked Autoencoders Are Scalable Vision Learners." ResearchGate, 2022. URL: https://www.researchgate.net/publication/353252838_Masked_Autoencoders_Are_Scalable_Vision_Learners



Міністерство освіти і науки України
Хмельницький національний університет



СЕРТИФІКАТ

Вусатий Нікіта Олександрович

учасник XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»
24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024

Олег СИНЮК

проректор Хмельницького національного
університету з наукової роботи,
доктор технічних наук, професор

м. Хмельницький
15-16 листопада 2024

E-mail: apkt.khnu@gmail.com

Додаток Д

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

МЕТОД ПІДВИЩЕННЯ ТОЧНОСТІ ВИЗНАЧЕННЯ ЗМІЩЕННЯ ОБ'ЄКТІВ ЗА
АНАЛІЗОМ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ МЕТОДІВ КЛАСТЕРИЗАЦІЇ

Виконавець: студент групи КНм-23-1 ВУСАТИЙ НІКІТА ОЛЕКСАНДРОВИЧ

Керівник: к.т.н., доцент кафедри КН ПАСІЧНИК ОЛЕКСАНДР АНАТОЛІЙОВИЧ

1



МЕТА

Метою роботи є підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

2

ЗАДАЧІ

- провести аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектувати метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначити набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконати програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначити точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.



3

АКТУАЛЬНІСТЬ

Інформаційні технології стали невід'ємною частиною нашого повсякденного життя, а аналіз зображень став ключовим компонентом у різних сферах. У відеоспостереження він допомагає стежити за дорожнім рухом, виявляючи рухомі транспортні засоби, оцінюючи їх швидкість і автоматично виписуючи штрафи за порушення. У медицині аналіз зображень життєво важливий для діагностики захворювань і планування операцій, демонструючи його важливість у сучасній охороні здоров'я. Крім того, робототехніка значною мірою покладається на точний аналіз зображень для таких завдань, як навігація та прибирання, а автоматизація підвищує ефективність повсякденних процесів, таких як електронні черги та пошук інформації. Загалом широке використання інформаційних технологій і аналізу зображень підкреслює їхню важливість у підвищенні зручності та функціональності багатьох програм.

4

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Процес визначення зміщення
об'єктів за аналізом зображень з
використанням методів
кластеризації

5

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи кластеризації

6

НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

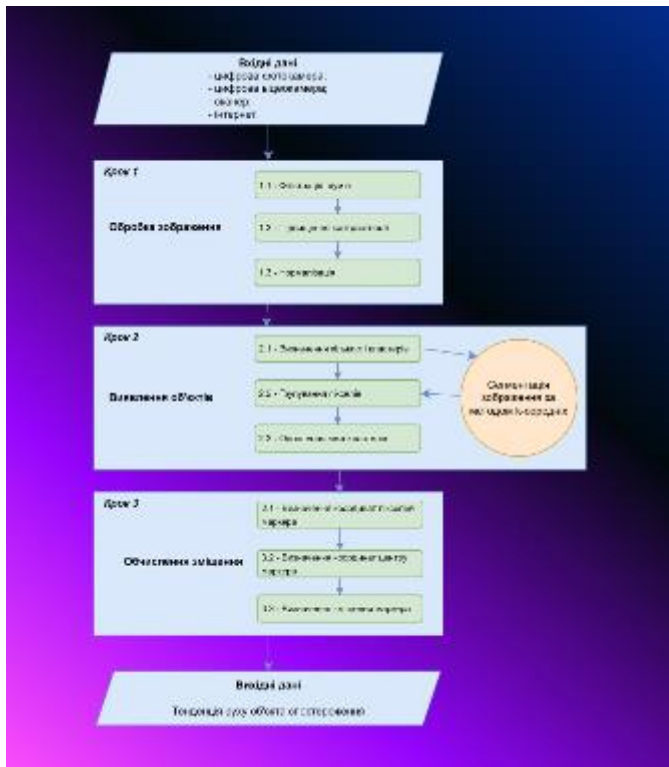
Удосконалено метод підвищення точності визначення зміщення об'єктів за аналізом зображень, з використанням методів кластеризації за рахунок використання кластеризації при визначенні контурів маркера на рухомому об'єкті

7

АПРОБАЦІЯ РЕЗУЛЬТАТІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ТА ПУБЛІКАЦІЇ

- Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція "Актуальні проблеми комп'ютерних наук (АПКН – 2024)", м. Хмельницький, ХНУ, 15-16 листопада 2024 р. (Вусатий Н.О., Пасічник О.А., Скрипник Т.К. Сегментація зображень // Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції "Актуальні проблеми комп'ютерних наук (АПКН – 2024)". – Хмельницький: ХНУ, 2024. – С. 112 – 114. <https://elar.khmnu.edu.ua/handle/123456789/171373>).
- За темою кваліфікаційної роботи магістра автором виконано 1 наукову публікацію Вусатий Н., Пасічник О., Манаюк Е., Скрипник Т. Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах // Вісник Хмельницького національного університету, №4, 2024 (339), С. 5963. DOI 10.31891/2307-5732-2024-339-4-10.

8

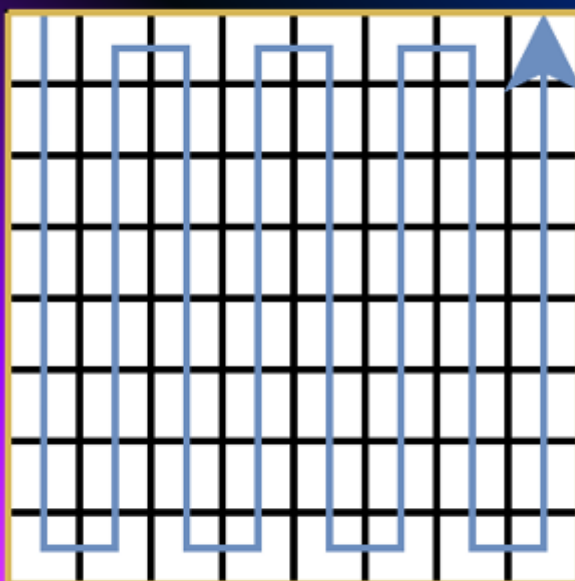


ЗАГАЛЬНА СХЕМА КРОКІВ ВИЗНАЧЕННЯ ЗМІЩЕННЯ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ МЕТОДУ КЛАСТЕРИЗАЦІЇ

Процес поділяється на три кроки:

1. Обробка зображення.
2. Виявлення об'єктів.
3. Обчислення зміщення.

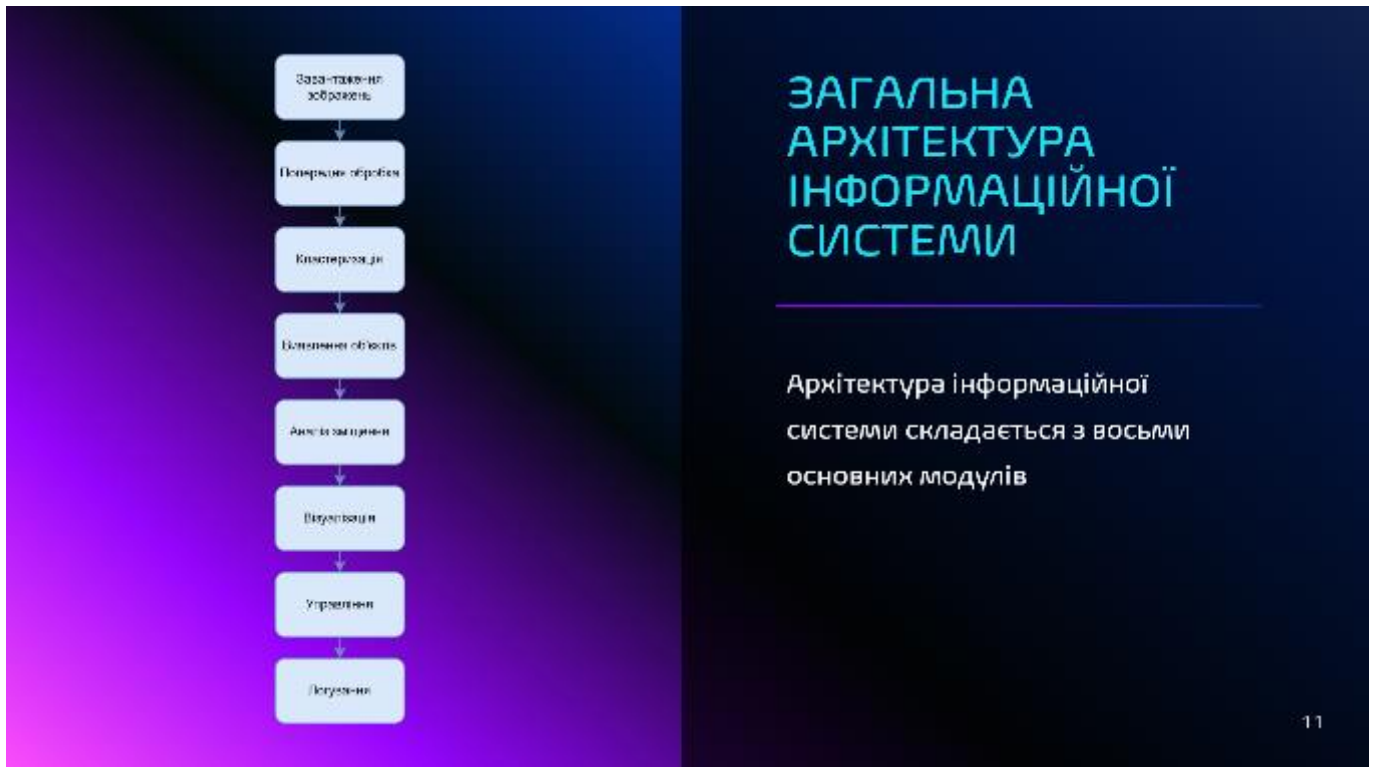
9



ГЕОМЕТРИЧНА ФІГУРА КВАДРАТА У ФОРМІ МАРКЕРА

Демонстрація обходу маркеру по пікселям

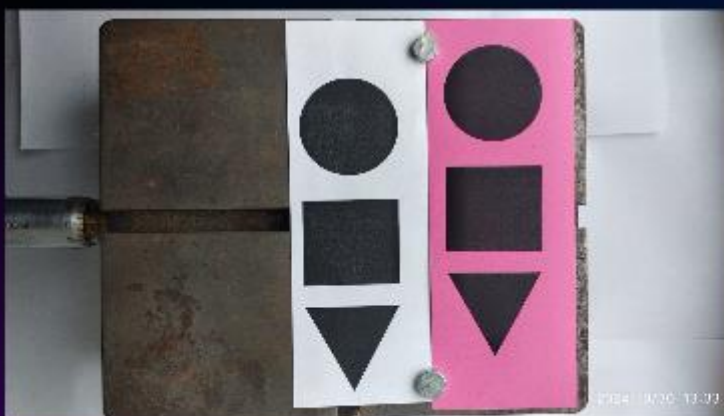
10



ДЕМОНСТРАЦІЯ ІНТЕРФЕЙСУ ПРОГРАМИ

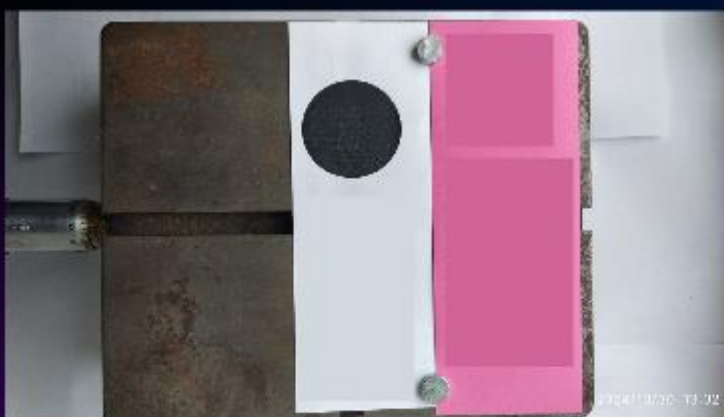
12

ПРИКЛАД ФОТО ДЛЯ ПРОВЕДЕННЯ ТЕСТІВ



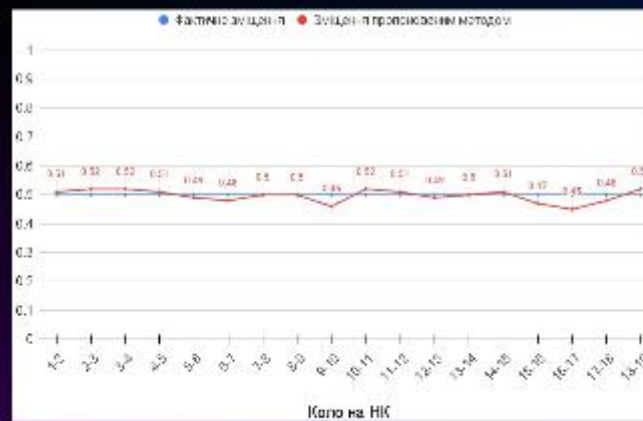
13

ПРИКЛАД РЕТУШУВАННЯ ЗАЙВИХ ГЕОМЕТРИЧНИХ ФІГУР ОКРІМ КОЛА НА ВИСОКО-КОНТРАСТНОМУ ФОНІ



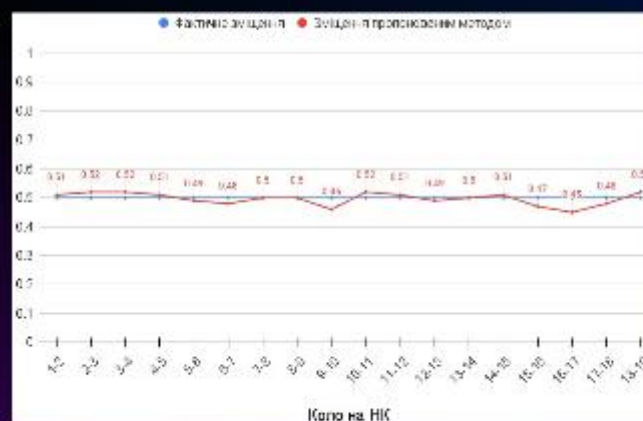
14

КОЛО НА НИЗЬКОМУ КОНТРАСТІ У ПРОПОНОВАНОМУ МЕТОДІ



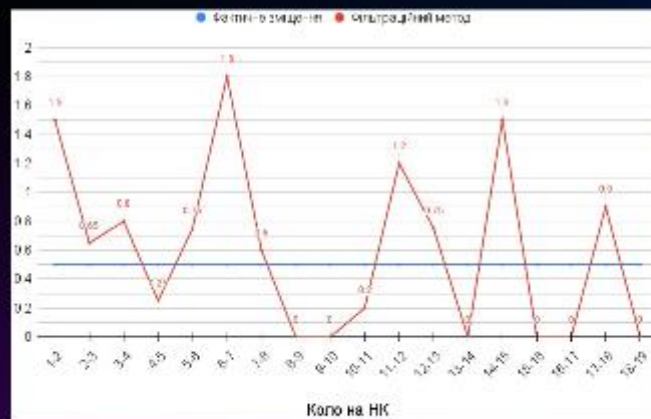
15

КОЛО НА НИЗЬКОМУ КОНТРАСТІ У ПРОПОНОВАНОМУ МЕТОДІ



16

КОЛО НА НИЗЬКОМУ КОНТРАСТІ У ФІЛЬТРАЦІЙНОМУ МЕТОДІ



17

МАТЕМАТИЧНЕ СПОДІВАННЯ



18

ДИСПЕРСІЯ



19

СЕРЕДНЄ КВАДРАТИЧНЕ ВІДХИЛЕННЯ



20

КОЕФІЦІЄНТ ВАРІАЦІЇ



21

СТУПІНЬ РОЗПІЗНАННЯ



22

ВИСНОВКИ

Під час виконанні кваліфікаційної роботи магістра було запропоновано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням кластеризації. Було розроблено метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням кластеризації. Підвищення точності визначення величини зміщення об'єкту на зображеннях досягається за рахунок використання методу кластеризації k -середніх при визначенні контурів маркеру, переміщення якого приймаються ідентичними переміщенням об'єкту спостереження.

23

ВИСНОВКИ

Побудовано загальну структуру методу підвищення точності визначення зміщення об'єктів з адаптацією та імплементацією методу k -середніх з покроковою його деталізацією. Сегментація зображень виконувалася з використанням методу k -середніх. Автоматизовано сегментацію зображень у поєднанні з фільтрацією шумів, підвищенням контрастності та нормалізацією.

24

ВИСНОВКИ

Було реалізовано поставлені задачі:

- проведено аналіз моделей та методів визначення зміщення об'єктів за аналізом зображень;
- спроектовано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- визначено набір критеріїв для оцінки результатів визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації;
- виконано програмну реалізацію методу підвищення точності визначення зміщення об'єктів;
- визначено точність визначення зміщення об'єктів запропонованим методом та навести порівняння з іншими відомими підходами.

25

ВИСНОВКИ

Виконано програмну реалізацію методу підвищення точності визначення зміщення об'єктів за допомогою кластеризації. Програмна реалізація виконана на основі модульній архітектурі системи, з 8 модулів, серед яких завантаження зображення, попередня обробка, кластеризація, виявлення об'єктів, аналіз зміщення, візуалізація, управління та логування. Програмна реалізація розроблена на мові програмування Python та за допомогою таких бібліотек як OpenCV, scikit-learn, Tkinter, PIL, skimage та інших. Виконано експериментальне тестування запропонованого методу на серії зображень для трьох форм маркерів чорного кольору, які штучним чином наносилися на висококонтрастний, що маркера, (білий) та низькоконтрастний (рожевий) фон поверхні.

26

ВИСНОВКИ

Експериментальна перевірка довела підвищення точності вимірювання переміщення за аналізом зображень з використанням методів кластеризації. Так відхилення середнього значення величини зміщення отримане під час експериментальної апробації методу від фактичного не перевищувало ~2%, в той час як для порівнюваного методу це ж відхилення не спостерігалось меншим за 20%. Значення середньоквадратичного відхилення для запропонованого методу мають значення одного порядку із величиною точності фізичного приладу, що використовувався під час практичного тестування методу, в той час як за умови невикористання методів кластеризації при визначенні контуру маркеру мають місце суттєво гірші показники.

27

ДЯКУЮ ЗА УВАГУ!

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 10%**

ID: 155969 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації Додано в БД: 2024-12-07 Автора: Нікіта ВУСАТИЙ Керівники: Олександр ПАСТІЧНИК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	66846	991	1694 (3%)	16 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Нікіта ВУСАТИЙ

Співавтор:

Назва: Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

Науковий керівник: Олександр ПАСТЧНИК, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 1.3%

Коефіцієнт подібності 2: 0.6%

Мікропробіли: 0

Заміна букв: 18

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-07 19:37:59.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата 7, 12, 2024

експерт

П. Петровський С.Р.

РІШЕННЯ ЕКСПЕРНОЇ КОМПІСІ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованою системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод підвищення точності визначення змінення об'єктів за аналізом зображень з використанням методів кластеризації

Автор: Вусатий Нікіта Олександрович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Пасічник Олександр Анатолійович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 2,0 %, схожість виявлена зі звітом автора з науково-дослідної практики.

2) за програмою StrikePlagiarism КПІ 1.3%, КЦ 0.12%,

які містять матеріали огляду предметної області; інші схожості є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи. Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається

Керівник роботи

Гарант ОП

Завідувач кафедри КН

Олександр ПАСІЧНИК

Руслан БАГРІЙ

Олександр БАРМАК



ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНм-23-1 ВУСАТОГО Нікіти за темою: Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

1. Актуальність теми

Сьогодні неможливо уявити світ без інформаційних технологій. Куди не подивись, вони всюди використовуються. Починаючи від простої, але важливої автоматизації якихось процесів і закінчуючи дійсно складними технологіями з важкими розрахункам. Аналіз зображення стає все більш популярним у різноманітних сферах діяльності, у тому числі й при виявленні зміщення об'єктів спостереження та при визначенні, за потреби, величини цього зміщення. Обрана тема є актуальною.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Кваліфікаційна робота магістра КНм-23-1 Нікіти Вусатого за ступенем обґрунтованості наукових положень, новизни, а також обсягом, структурою та змістом викладеного матеріалу відповідає вимогам щодо наукових робіт. У роботі використані методи, що повністю відповідає предметній області спеціальності 122 Комп'ютерні науки

3. Професійні та особистісні якості магістранта

В період виконання кваліфікаційної роботи магістра Нікіта Вусатий виявив себе кваліфікованим фахівцем здатним на високому рівні виконувати поставлені завдання. Володіє необхідними професійними навичками та загальними компетентостями.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Кваліфікаційна робота виконана студентом особисто. Визначення мети та постановка задач виконувалося спільно з науковим керівником.

5. Наукова новизна та оригінальність запропонованих підходів

Удосконалено метод підвищення точності визначення зміщення об'єктів за аналізом зображень за рахунок використання кластеризації при визначенні контурів маркера на рухомому об'єкті. Результати роботи доповідалися на XVI Всеукраїнській науково-практичній конференції "Актуальні проблеми комп'ютерних наук (АПКН –

20243)”, м. Хмельницький, ХНУ, 15-16 листопада 2024 р. та були опубліковані у Віснику Хмельницького національного університету, №4, 2024 (339), С. 59 – 63.

6. Ступінь оволодіння методами дослідження

Продемонстровано високий рівень володіння методами дослідження, які були використанні у роботі.

7. Повнота та якість розкриття теми роботи

Тема роботи розкрита якісно на високому рівні, задачі дослідження виконані в повному обсязі.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Позитивними рисами кваліфікаційної роботи є системність та послідовність викладення матеріалу. Продемонстрована здатність збирати і аналізувати дані, для забезпечення якості прийняття рішень. У кваліфікаційній роботі магістра формалізовані та систематизовані вимоги до розробленої комп'ютерної системи. Робота відповідає всім граматичним нормам та демонструє зрозумілий, виважений стиль подання інформації.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Результати роботи можуть бути використані для широкого кола завдань зі спостереження за рухомими об'єктами та визначенням величини їх зміщення.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Кваліфікаційна робота магістра Нікіти Вусатого виконана повністю у відповідності із представленими вимогами та є завершеною науковою працею. Вона містить рішення наукової задачі, яка по суті полягає у реалізації методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. З огляду на вище сказане, робота рекомендується до захисту та заслуговує на оцінку «відмінно».

Науковий керівник _____



к.т.н., доц., Олександр ПАСІЧНИК



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

ср. КНм-23-1 ВУСАТОГО Нікіти за темою: Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації

1. Актуальність обраної теми

Сьогодні неможливо уявити світ без інформаційних технологій. Куди не подивись, вони всюди використовуються. Починаючи від простої, але важливої автоматизації якогось процесів і закінчуючи дійсно складними технологіями з важкими розрахункам. Аналіз зображення стає все більш популярним у різноманітних сферах діяльності, у тому числі й при виявленні зміщення об'єктів спостереження та при визначенні, за потреби, величини цього зміщення. Обрана тема є актуальною.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи у повній мірі відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра згідно Стандарту освіти

3. Повнота розкриття мети та завдань дослідження

Було проведено порівняння та аналіз можливих методів розв'язання поставленої задачі та обрано сучасний підхід щодо обробки зображень в частині підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. Також було досліджено існуючі реалізації розв'язання подібних задач. В рамках кваліфікаційної роботи магістра було проведено аналіз предметної області, розроблено метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. Виходячи з наведених положень, мета є повністю розкритою, а завдання дослідження виконані в повному обсязі.

4. Наявність наукової новизни

В результаті проведення кваліфікаційної роботи були отримані наступні результати, а саме - удосконалено метод підвищення точності визначення зміщення об'єктів за аналізом зображень за рахунок використання кластеризації при визначенні контурів маркера на рухомому об'єкті. Результати роботи доповідалися на XVI Всеукраїнській науково-практичній конференції "Актуальні проблеми комп'ютерних наук (АПКН – 2024)", м. Хмельницький, ХНУ, 15-16 листопада 2024 р. та були опубліковані у Віснику Хмельницького національного університету, №4, 2024 (339), С. 59 – 63.

5. Зміст кожного розділу роботи

В першому розділі наведено характеристику предметної області з оглядом методів аналізу зображень для визначення положення об'єктів, оглядом методів кластеризації для аналізу зображень та аналізом існуючих підходів до підвищення точності вимірювань у технічних системах. Визначено мету та задачі дослідження. В другому розділі реалізовано метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. В третьому розділі виконана програмна реалізація методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації та виконано тестування програми. В четвертому розділі виконано дослідження реалізованого методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації.

6. Ступінь розкриття теми роботи

Кваліфікаційна робота магістра присвячена методу підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації. В рамках роботи було проаналізовано предметну область, програмні системи для вирішення аналогічних завдань та створено відповідний метод та його програмна реалізація із проведенням необхідного тестування та досліджень. Тема роботи розкрита якісно на високому рівні, задачі дослідження виконані в повному обсязі.

7. Якість оформлення кваліфікаційної роботи

Кваліфікаційна робота магістра відповідає всім вимогам до оформлення таких робіт. Стиль подання інформації є фаховим та зрозумілим. Робота не містить стилістичних відхилень та відповідає всім нормам граматики. Робота виконана логічно, послідовно та аргументовано. Матеріал викладено якісно із дотриманням вимог до професійного літературного стилю.

8. Недоліки кваліфікаційної роботи

В роботі не зазначено, чим обумовлено використання маркерів зазначеної форми

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Беручи до уваги новизну, актуальність, важливість отриманих результатів, їх достовірність та обґрунтованість, вважаю, що кваліфікаційна робота магістра Нікіти Вусатого «Метод підвищення точності визначення зміщення об'єктів за аналізом зображень з використанням методів кластеризації» є оригінальним та завершеним науковим дослідженням. Кваліфікаційна робота магістра Нікіти Вусатого рекомендується до захисту, рекомендована оцінка «відмінно».

Опонент (прізвище, ім'я, по батькові, посада, місце роботи)

Лисенко Сергій Михайлович, доктор тех. наук, професор КТ
ХНУ

«09» грудня 2024 р.

Підпис